What Makes the Preferred Thinking Direction for LLMs in Multiple-choice Questions?

Yizhe Zhang[1]*, **Richard He Bai**[1]*, **Zijin Gu**[1]†, **Ruixiang Zhang**[1], **Jiatao Gu**[1], **Emmanuel Abbe**[1], **Samy Bengio**[1], **Navdeep Jaitly**[1]

[1]Apple
{yizhe_zhang, richardbai, njaitly}@apple.com

Abstract

Language models usually use left-to-right (L2R) autoregressive factorization. However, L2R factorization may not always be the best inductive bias. Therefore, we investigate whether alternative factorizations of the text distribution could be beneficial in some tasks. We investigate right-to-left (R2L) training as a compelling alternative, focusing on multiple-choice questions (MCQs) as a test bed for knowledge extraction and reasoning. Through extensive experiments across various model sizes (2B-8B parameters) and training datasets, we find that R2L models can significantly outperform L2R models on several MCQ benchmarks, including logical reasoning, commonsense understanding, and truthfulness assessment tasks. Our analysis reveals that this performance difference may be fundamentally linked to multiple factors including calibration, computability, and directional conditional entropy. We analyze the impact of these factors through controlled simulation studies using arithmetic tasks, where the impacting factors can be better disentangled. Our work demonstrates that exploring alternative factorizations of the text distribution can lead to improvements in LLM capabilities and provides theoretical insights into optimal factorization towards approximating human language distribution, and when each reasoning order might be more advantageous. Our code and checkpoints are released at https://github.com/apple/ml-reversal-blessing

1 Introduction

Large Language Model (LLM) pretraining commonly employs left-to-right (L2R) next-token prediction, an approach that enables efficient parallelization and caching. This method models the text distribution $p(x)$ as a factorized autoregressive chain as $p(x_t|x_{<t})$. L2R naturally aligns with human cognitive processes of text generation and reasoning, making it well-suited for inference tasks.

However, while perfect modeling of each $p(x_t|x_{<t})$ would theoretically enable exact recovery of the data distribution $p(x)$, neural networks inevitably introduce approximation errors for each $p(x_t|x_{<t})$. These errors compound over timestep t during inference, potentially resulting in hallucinations and repetitions in generation (Bengio et al., 2015; Zhang et al., 2023). Further, L2R factorization can result in inductive biases that lead to unwanted behaviors. For example, Allen-Zhu & Li (2023a) show that inverse search is challenging for L2R LLMs, and Berglund et al. (2023) demonstrate the "reversal curse" where models trained on forward text data struggle with inverse relationships.

[*] Equal contribution.
[†] Core contribution.

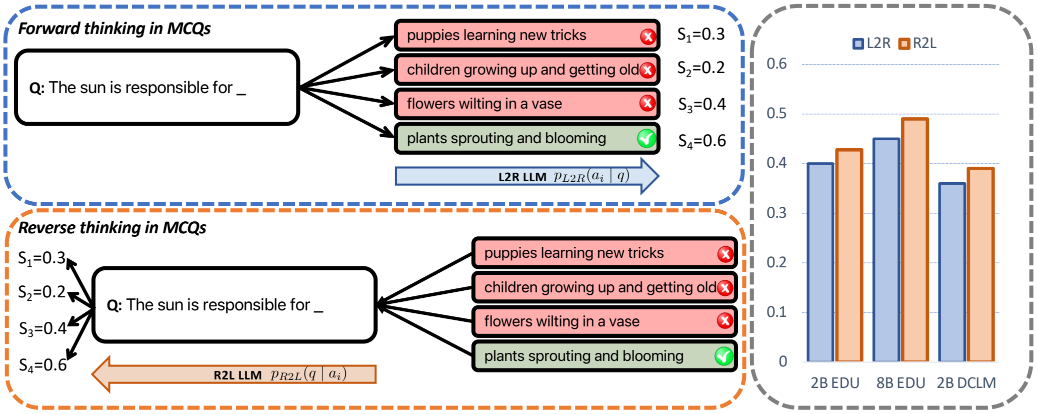
Forward thinking in MCQs
Q: The sun is responsible for _
puppies learning new tricks
children growing up and getting old
flowers wilting in a vase
plants sprouting and blooming
S_1=0.3
S_2=0.2
S_3=0.4
S_4=0.6
L2R LLM $p_{L2R}(a_i \mid q)$
Reverse thinking in MCQs
S_1=0.3
S_2=0.2
S_3=0.4
S_4=0.6
Q: The sun is responsible for _
puppies learning new tricks
children growing up and getting old
flowers wilting in a vase
plants sprouting and blooming
R2L LLM $p_{R2L}(q \mid a_i)$
L2R
R2L
0.6
0.5
0.4
0.3
0.2
0.1
0
2B EDU
8B EDU
2B DCLM

Figure 1: Reverse Thinking in MCQs. **Top left**: Standard forward thinking evaluates each answer choice based on the question and selects the one with the highest relevance score in a L2R LLM. **Bottom left**: Reverse thinking evaluates the question based on each answer choice and selects the answer that maximizes the relevance score in a R2L LLM. **Right**: Reverse thinking consistently outperforms forward thinking in certain MCQ tasks (Openbook QA in this figure), independent of training data and model size.

We investigate whether L2R is optimal, and if alternative factorizations might capture unique aspects of the data distribution that complement L2R. Can specific factorizations achieve lower approximation errors compared to L2R, or reduce L2R's inherent bias in particular task domains?

Autoregressive modeling in right-to-left (R2L) fashion factorizes $p(x)$ as $p(x_t|x_{>t})$, which presents a particularly promising alternative that has been examined in previous work (Papadopoulos et al., 2024; Berglund et al., 2023; Zhang-Li et al., 2024). This setup views the task as predicting the previous token, and it can achieve prediction losses comparable to the L2R next token prediction objective, due to its symmetry to L2R. While R2L may seem counterintuitive given human language processing patterns, it may enable more efficient knowledge extraction in certain scenarios by aligning with the natural direction of information flow in those cases, and it could provide complementary inductive biases that help with specific reasoning tasks.

We investigate three questions: (1) How to evaluate R2L models on knowledge extraction and basic reasoning tasks? (2) Can R2L factorization match or surpass L2R's capabilities in knowledge extraction and reasoning for downstream tasks? (3) What are the underlying factors determining the preference of L2R or R2L factorizations?

To address these questions, we conducted controlled experiments comparing L2R and R2L models trained with identical data and computational resources. We evaluated both factorization approaches using standard LLM benchmarks with Multiple-Choice Questions (MCQs). For simplicity, we limit our comparison to MCQs, and leave the evaluations for generative tasks as future work. For R2L models, we applied Bayesian inference to implement "reverse thinking," evaluating choices based on their likelihood of generating the prompt (Figure 1).

Our results reveal a surprising and previously unobserved empirical finding: R2L models can significantly outperform L2R models on several standard MCQ benchmarks, including tasks requiring logical reasoning, commonsense understanding, and truthfulness assessment. This finding—where backward thinking can sometimes yield superior performance—is a phenomenon not previously observed or widely recognized in the current literature, especially at the scale and across the diverse benchmarks we evaluated. This observation challenges the prevailing fundamental assumption that L2R autoregressive factorization is the default and optimal inductive bias for LLMs.

Beyond this empirical discovery, our work introduces new perspectives on analyzing the performance differences between L2R and R2L models by proposing three potential underlying factors: *calibration*, *computability*, and, critically, *conditional entropy*. The role of conditional entropy in explaining the preferred reasoning direction, and generally understanding reasoning machinery, is a novel theoretical insight introduced in this paper. We empirically verify this hypothesis, showing that lower conditional entropy generally correlates with higher accuracy in the reasoning direction for many tasks. Our primary contribution lies not in demonstrating that one factorization is universally superior, but in developing a principled framework for understanding when and why different factorization directions may be preferred.

Nevertheless, these factors are intricately interwoven in actual MCQs, complicating the analysis. To disentangle these factors and ablate on their impact to the performance of L2R or R2L factorization, we design a controlled simulation study using arithmetic tasks, revealing how various factors influence the effectiveness of certain factorization. Our code and model checkpoints have been made publicly available for reproduction and to facilitate future research. [3]

2 Thinking Backward in MCQs

2.1 Solving MCQs

Solving MCQs with forward thinking As shown in Figure 1, in MCQs, LLMs process a question q alongside a set of answer choices $A = \{a_1, a_2, \ldots, a_n\}$. Each (question, answer) pair (q, a_i) is encoded to compute a relevance score s_i. The model then selects the answer a_k corresponding to the highest score: $k = \arg\max_i s_i$.

To compute s_i, the model evaluates the log-probability of generating the answer a_i given the question q. This log-probability is often normalized to account for variations in answer length, preventing a bias toward shorter or longer responses. Various normalization techniques (Holtzman et al., 2021) can be applied, however, we resort to the most common approach which divides the total log-probability by the length of the answer $N_i = \text{len}(a_i)$ in tokens or bytes, resulting in a normalized relevance score: $s_i = \frac{\log p(a_i|q)}{N_i}$. The log-probability is factorized as

$$\log p(a_i \mid q) = \sum_{l=1}^{N_i} \log p_{L2R}(a_i^l \mid q, a_i^{<l}), \tag{1}$$

where a_i^l represents the l-th token in a_i.

Solving MCQs with reverse thinking With an R2L model, s_i can be computed using Bayes' rule:

$$s_i = \log p(a_i \mid q)/M_i = \frac{1}{M_i}(\log p_{R2L}(q \mid a_i) + \log p_{R2L}(a_i) - C),$$

where $M_i = \text{len}(q, a_i)$, $C = \log p_{R2L}(q)$ is a constant. $\log p_{R2L}(q \mid a_i)$ and $\log p_{R2L}(a_i)$ can be autoregressively factorized in R2L manner similar to the forward thinking process in Eq. equation 1. We consider 3 paradigms of the s_i for reverse thinking: (1) normalized s_i with $M_i = \text{len}(q, a_i)$ resembling the forward thinking; (2) unnormalized s_i with $M_i = 1$; (3) unnormalized s_i without prior, i.e. $s_i = \log p_{R2L}(q \mid a_i)$.

2.2 Model evaluation

We conducted our evaluation on standard LLM evaluation tasks with MCQs that cover different domains including commonsense reasoning, logical reasoning, truthfulness evaluation and more.

Our evaluation tasks include HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021), Openbook QA (Mihaylov et al., 2018), MathQA (Amini et al., 2019), LogiQA (Liu et al., 2020), PIQA (Bisk et al., 2019), Social IQA (Sap et al., 2019), Commonsense QA (Talmor et al., 2018), Truthful QA (Lin et al., 2021), and WinoGrande (Sakaguchi et al., 2021). For ARC (easy, hard) and MMLU, we combine all the subtasks to report the overall score. We use Eleuther-AI LM-eval harness (Gao et al., 2024) for all the evaluations. For MMLU, LogiQA, and Commonsense QA, we modify the task templates to present full answer choices rather than just choice labels.

2.3 Model Pretraining

To pretrain the model, we first tokenize each complete dataset. The R2L model is then trained by reversing all tokens within each training data instance. For a fair comparison between the R2L and

[3] https://github.com/apple/ml-reversal-blessing.

Table 1: Comparing L2R and R2L on MCQs. All the models are trained on 350B non-repeating tokens. The HF-2B baseline is from Penedo et al. (2024). We directly used their reported numbers. EDU-2B, EDU-8B and HF-2B models are trained with the same FineWeb-EDU 350B dataset. Green indicates R2L wins, red indicates R2L loses.

	DCLM-2B			**EDU-2B**			**EDU-8B**			**HF-2B**
	L2R	R2L	% Change	L2R	R2L	% Change	L2R	R2L	% Change	L2R
Training loss	**2.668**	2.724	+2.10	**2.345**	2.396	+2.17	**2.087**	2.138	+2.44	-
LogiQA	30.57	**31.64**	+3.52	27.96	**31.49**	+12.64	29.95	**31.03**	+3.61	-
OpenbookQA	36.00	**38.40**	+6.67	42.40	**44.40**	+4.72	45.00	**48.40**	+7.56	41.04
TruthfulQA	19.82	**29.99**	+51.23	24.36	**28.76**	+18.09	24.97	**31.70**	+26.95	-
CommonsenseQA	42.83	**45.29**	+5.74	42.92	**45.13**	+5.15	39.15	**44.96**	+14.84	36.60
Social IQA	**41.56**	40.94	-1.48	**42.78**	42.22	-1.32	**44.58**	43.50	-2.42	40.52
ARC	**54.11**	43.88	-18.91	**60.65**	52.31	-13.75	**68.29**	56.22	-17.67	57.47
HellaSwag	**60.87**	45.89	-24.62	**60.57**	44.34	-26.79	**71.60**	49.22	-31.26	59.34
MathQA	**26.50**	22.21	-16.18	**26.80**	24.86	-7.25	**28.77**	25.33	-11.96	-
MMLU	**31.66**	31.31	-1.10	**34.57**	34.35	-0.62	**38.90**	37.11	-4.60	37.35
PIQA	**74.43**	58.05	-22.00	**74.48**	57.13	-23.30	**77.80**	59.14	-23.98	76.70
Winogrande	**61.01**	53.51	-12.29	**60.93**	54.85	-9.97	**65.75**	54.70	-16.81	57.54

L2R models, both models are pretrained from scratch using the same Fineweb-EDU subset dataset comprising 350B tokens (Penedo et al., 2024). Each model consists of 2B parameters (**EDU-2B**), which is the default setting in our experiments. We also train 1.5B, 4B, 8B L2R and R2L models with the same 350B Fineweb-EDU dataset, and 2B L2R and R2L models trained with a random subset of the DCLM dataset (Li et al., 2024a) containing 350B tokens (**DCLM-2B**). Both the L2R and R2L models are trained for a single epoch, ensuring each training instance is seen only once, thus the training loss should align with the validation loss. More details for model architecture and training are provided in Appendix A.

2.4 Results

We present our results in Table 1. To verify our pretraining pipeline, we first compare the performance of our pretrained model with the 2B model trained by Huggingface (Penedo et al., 2024) (**HF-2B**) [4]. Under similar model size and the same dataset, our 2B model (**EDU-2B**) achieves performance comparable to or exceeding the L2R results reported by Huggingface **HF-2B**. Full results for all models ranging from 1.5B to 8B parameters are provided in our appendix B.

We compared L2R and R2L model performance across all evaluated tasks, employing bootstrap sampling (5 replicates, each with 80% resampling with replacement) for statistical robustness. As shown in Table 1, R2L models with reverse thinking exhibited significantly better reasoning performance on 4 out of 11 tasks: LogiQA, OpenBookQA, TruthfulQA, and CommonsenseQA. Statistical significance results are presented in Table 6.

These results remained consistent across different model sizes (1.5B to 8B), datasets (DCLM, FineWeb EDU), and random seeds, indicating the findings are not due to random fluctuation. The relative performance gain or loss when switching to R2L remained generally stable as model size increased (see appendix B).

For TruthfulQA specifically, we observed the most significant performance gain with R2L the improvement was substantial (51.23% on DCLM-2B). We hypothesize that the "reverse thinking" may inherently align better with truthfulness assessment, as it evaluates the question based on each answer choice rather than generating answers from the question. This framing might help the R2L model better discern subtle inaccuracies that an L2R model might overlook due to "surface form competition". Additionally, R2L models demonstrate significantly lower conditional entropy on TruthfulQA compared to L2R models, which aligns with our hypothesis that lower conditional entropy is associated with higher task accuracy.

For reverse thinking with R2L, we use the paradigm 3 (*i.e.*, unnormalized s_i without prior) for downstream tasks evaluation. We compare the three paradigms for reverse thinking in Appendix C, Table 5. Ideally, s_i should incorporate priors, as in paradigm 1 or 2. However, in practice, using s_i without prior (paradigm 3) consistently yields the best performance except for Social IQA and

[4] https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

PIQA. We hypothesize this may be due to intrinsic difficulty of estimating the prior probabilities $p(a)$ using LLMs, due to the "surface competition" calibration issues (Holtzman et al., 2021). We provide detailed explanation of our hypothesize using an illustrative example in Appendix E.

Intuitively, paradigm 3 which uses $p(q \mid a)$ is sensible. In MCQs, answer choices are typically well-formed and reasonable text, meaning their prior probabilities $p(a)$ are unlikely to vary significantly among choices, assigning a uniform prior is probably a reasonable approach. Consequently, $p(a \mid q)$ and $p(q \mid a)$ tend to be highly correlated. Consider a real-world example from Openbook QA: for the question (q) "A magnet will stick to", candidate answers (a_i) include "a belt buckle", "a wooden table", "a plastic cup", and "a paper plate". A model can deduce that "a belt buckle" is far more likely to be associated with the question "A magnet will stick to" compared to the other options, demonstrating how $p(q \mid a)$ can effectively capture the relevance between question and answer.

We also monitor the training loss for pretraining the models on both directions. We observed findings similar to Papadopoulos et al. (2024) in that L2R yields a lower loss compared to R2L, even though both models model the same target data distribution. In Papadopoulos et al. (2024), the largest model that was trained had 405M parameters while our models were trained at the popular small LLM size range of 2B-8B parameters. At this size, we observe a similar percentage difference as reported by previous work, of about 2%-2.5% increase in loss when using R2L, indicating learning the R2L factorization is more challenging. This makes it particularly interesting that on a bunch of MCQ tasks we see the R2L is performing better, as elaborated above.

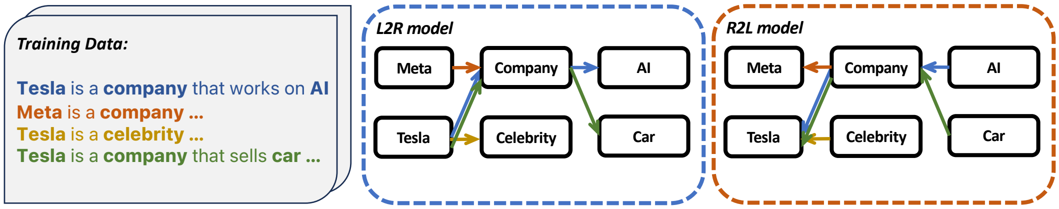

Figure 2: L2R and R2L LLMs pretrained on the same data will generate opposite search graphs based on the order in which they process the information entities.

3 What Makes the Preferred Order of Thinking?

We then seek to gain a deeper understanding of why there is a preferred orientation for the MCQs. We explore three main hypotheses (3**C**): ***C**alibration*, ***C**omputability*, and ***C**onditional entropy*. Admittedly, there may be other factors that we have overlooked that contributes to this preference.

3.1 Calibration

The first potential explanation concerns the scoring mechanism in forward thinking, where $s_i = \log p_{L2R}(a_i \mid q)$. Eq. equation 1 might not lead to an optimal estimation of $p(a|q)$ as it suffers from several calibration issues. Among the choices, some may contain more words that are highly predictable (e.g., "Hong Kong" or stop-words like "a"), potentially leading to spuriously inflated relevance scores. Additionally, Holtzman et al. (2021) shows that simple probability normalization in MCQs is challenging because different surface forms of semantically equivalent answers compete for probability mass, potentially *diluting* scores for correct answers due to this "surface form competition".

In contrast, reverse thinking with paradigm 3, where $s_i = \log p_{R2L}(q \mid a_i)$, mitigates this issue since the target question q remains constant across all choices. We provide rationale analysis on how R2L paradigm 3 alleviates "surface competition" in Appendix E. In a nutshell, forward thinking suffers from surface form competition, where semantically similar words (e.g., "dog" and "puppy") split probability mass, reducing the likelihood of selecting the correct answer. Reverse thinking mitigates this by enforcing a uniform prior, eliminating competition in the prior distribution and allowing a fairer comparison between answer choices. This suggests that reverse thinking inherently "auto-normalizes" different choices, resulting in more robust evaluation. However, this sole theory fails to explain why reverse thinking does not consistently outperform forward thinking across all tasks, instead showing superior performance only in specific MCQ scenarios.

3.2 Computability

A second potential theoretical explanation, which echoes with Papadopoulos et al. (2024), suggests that computational complexity may underlie these directional preferences. Drawing an analogy to number theory, where multiplying prime numbers is computationally straightforward, while the reverse operation of prime factorization is NP-hard.

It is tempting to consider this computational complexity asymmetry as the main underlying cause for why L2R or R2L is preferred for specific tasks. However, recent research (Mirzadeh et al., 2024; Kambhampati, 2024; Valmeekam et al., 2024) finds that LLMs may not actually perform genuine reasoning or computing, as evidenced by their poor generalization when tasks undergo minor modifications. This implies that LLMs mainly emulate *reasoning patterns* from their training data instead of carrying out actual logical computation, weakening the hypothesis that directional preferences stem from varying computability in different directions. Furthermore, most MCQs primarily involve knowledge retrieval and basic reasoning, which might not reach the complexity threshold where computational hardness would become a significant factor. Therefore, acknowledging that computability may be a factor, we keep exploring alternative hypotheses.

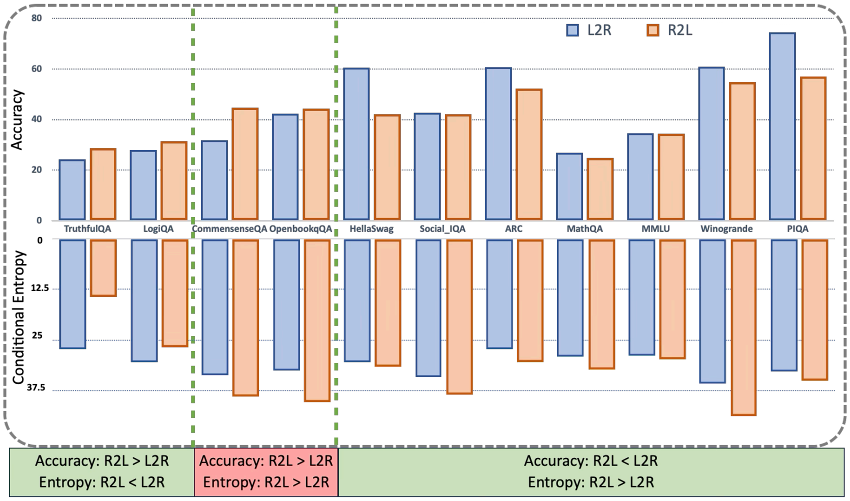

Figure 3: Lower conditional entropy may associate with higher accuracy in the reasoning direction.

3.3 Conditional Entropy

Our final hypothesis posits that the optimal direction of thinking is closely related to the *conditional entropy* of the downstream task. Recent work has shown that learning knowledge extraction and simple multihop reasoning is more challenging for problems with higher degree of **branching factors** or "**globality degree**" compared to those with lower branching factors and more deterministic relationships (Abbe et al., 2024). It is conceivable that directionality of data can impact the branching degree and lead to different learning efficiencies in different directions (for example multiplication in left-to-right direction is factorization in the opposite direction, each with different branching factors).

Previous work (Berglund et al., 2023; Allen-Zhu & Li, 2023b) has also demonstrated that LLMs suffer from the "reversal curse", indicating that inverse R2L search in LLMs is inherently challenging for L2R models - due the disconnect between training and inference directions. Consider an LLM trained on sequences of knowledge/information name entities $(e_1, e_2, \cdots, e_n)$. LLM may effectively construct a **directed** search graph that maps the key $(e_1, \cdots, e_{i-1})$ to the value e_i for any i. Following this logic, the training data essentially forms a Bayesian network that can be represented as a *directed acyclic graph* (DAG) of entities. Similarly, training an R2L model yields an analogous DAG but with reversed edge directions (see Figure 2 for an illustration). The search efficiency between these two graphs may vary given different queries.

We hypothesize here that for two different factorizations of the data, **the direction yielding lower conditional entropy will perform better in MCQs**, as it reflects better efficiency in knowledge extraction and multi-hop search. We note however, that this is only true when models under both factorization directions have sufficiently low error, which seems to be true for our models here.

More formally, for a downstream MCQ task T with question and answer choices following task-specific data distribution $P_T(q, a)$, we compare the *conditional entropy* in both directions under pretrained L2R and R2L models (Eq. equation 2 for L2R and Eq. equation 3 for R2L):

$$-\mathbb{E}_{q' \sim P_T(q)} \sum_a p_{L2R}(a|q') \log p_{L2R}(a|q'). \tag{2}$$

$$-\mathbb{E}_{a' \sim P_T(a)} \sum_q p_{R2L}(q|a') \log p_{R2L}(q|a'). \tag{3}$$

We assume that the conditional entropy is a proxy for the quality of the learned model, and the direction with lower conditional entropy should perform better. However, computing these summations in equation 2 and equation 3 is intractable due to the exponentially large candidate space. Therefore, we employ Monte Carlo estimation of equation 2 and equation 3 as proxy measures, specifically computing

$$-\mathbb{E}_{q' \sim P_t(q), a' \sim p_{L2R}(a|q')} \log p_{L2R}(a'|q'), \tag{4}$$

$$-\mathbb{E}_{a' \sim P_t(a), q' \sim p_{R2L}(q|a')} \log p_{R2L}(q'|a'). \tag{5}$$

Because of the extensive amount of evaluation datasets, due to limited computation budget, we only conducted a single sample rollout for $a' \sim p_{L2R}(a|q')$ and $q' \sim p_{R2L}(q|a')$. We recognize that this may not be a precise representation of the true conditional entropy, given that the candidate space grows exponentially with the maximum sequence length.

Empirical Verification To verify this hypothesis, we estimate the conditional entropy for all the evaluation tasks. We provide more experimental details in Appendix D. Figure 3 presents our empirical results, which support this hypothesis that lower conditional entropy is typically linked to greater task accuracy, except for CommonsenseQA and OpenbookQA which are outliers likely because of other confounding factors including the computability. In Figure 3, we observed that the conditional entropy of R2L is generally greater than L2R. This trend could be related to the findings presented in Table 1, indicating that R2L tends to have higher training loss too. Complementing the rationale in Papadopoulos et al. (2024), we hypothesize that the ease with which the language model can approximate the factorized distribution of L2R and R2L, may be also tied to which direction exhibits higher branching factors in that direction. We leave this exploration for future study.

Table 2: Results of the controlled simulation study of 4-digits multiplication. Theoretical Conditional Entropy (Theo. Cond. Ent.) represents the expected conditional entropy under an ideal model. L2R consistently outperforms R2L in Forward X, while R2L is superior in Reverse X. Lower conditional entropy correlates with higher accuracy.

	Forward X			**Reverse X**		
	L2R	R2L(m,n)	R2L(m)	R2L	L2R(m,n)	L2R(n)
Test Accuracy (%)	**99.81**±0.15	59.71±1.99	60.93 ± 0.88	**100**±0	97.82±0.35	99.85±0.10
Train Accuracy (%)	**99.76**±0.15	59.03 ± 1.66	61.22±1.12	**100**±0	97.90±0.42	99.98±0.04
Test Cond. Ent. (nats)	0.06	1.18	0.08	0	0.84	0.01
Train Cond. Ent. (nats)	0.06	1.17	0.08	0	0.83	0.01
Theo. Cond. Ent. (nats)	0	1.49	0	0	1.49	0
Training loss	**0.86**	0.94	0.94	**0.86**	0.94	0.94

4 Controlled Simulation Study

The three hypotheses discussed in Section 3 are intricately entwined in actual MCQs, making it challenging to disentangle them. To better investigate the hypotheses explaining the optimal direction for MCQs, we conducted a meticulously controlled simulation study (Figure 4) focused on 4-digit multiplication. Although the arithmetic dataset is different from the real language modeling datasets, this simulation study can be a good controlled experiment to understand the phenomenon we observed in Section 3, as we are investigating the underlying principle of the model with different factorizations regardless of the dataset. The L2R and R2L models were initialized **from scratch** and **exclusively** trained on this simulation dataset to eliminate any potential confounding factors. All data instances share the same format and length, removing the *calibration* effect from the analysis and allowing us to concentrate on *computability* and *conditional entropy*.

Experiment Setup We conduct two types of simulation experiments: Forward Multiplication (**Forward X**) and Reverse Multiplication (**Reverse X**). In Forward X, each training instance was represented as $m \times n = p$, where $m, n \in \{0, \ldots, 10^4\}$ and $p \in \{0, \ldots, 10^8\}$. The formatting included spaces between digits and mathematical operators to ensure a consistent single tokenization for both L2R and R2L models. In Reverse X, the multiplication was in reverse order, such as $p = m \times n$. For each simulation type, L2R and R2L models were trained with a 2B model size with 1 epoch on all 10^8 non-repeating equations except 1,000 test examples, totaling almost 3.2B tokens.

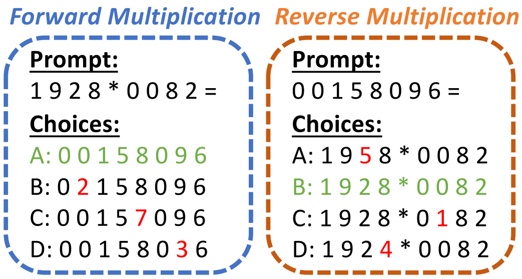

Figure 4: Simulation Study. Forward multiplication simulates a **many-to-one** mapping scenario, while reverse multiplication simulates a **one-to-many** mapping.

The model performance was assessed using the held-out test set with 1,000 examples. These examples were converted into a multiple-choice format consisting of 4 choices (Figure 4). Other than the correct answer, the remaining three hard negative options were created by altering a single digit in the correct answer to a random other digit, at a random position. The presenting order of the four choices are then randomly shuffled. We augmented the test set 10 times and calculated the average accuracy and conditional entropy.

As multiple pairs of m and n can be mapped to the same product p, Forward X is a **many-to-one** mapping. The theoretical conditional entropy for predicting the correct p from $m \times n$ is 0 under an oracle model. However, as there are several paths from the product p to the m, n pairs, the theoretical conditional entropy for predicting the $m \times n$ from p becomes 1.49 nats under an oracle model. In the Reverse X task, which transitions into a **one-to-many** scenario, the analysis is inverted.

For Forward X, we explore an alternative R2L evaluation method, denoted as **R2L(m)**, where the relevance score of the i-th choice p_i is calculated as $s_i = \log p_{R2L}(m \mid p_i, n)$, focusing on the conditional entropy of m rather than $m \times n$ as in the standard **R2L(m,n)** method. Since R2L(m) is essentially division, it is deterministic with a theoretical conditional entropy of 0. Similarly, we have a variant for L2R in reverse X, called **L2R(n)**.

Results The results are presented in Table 2. In Forward X scenarios, L2R models demonstrate higher accuracies than R2L(m,n) models, with correspondingly lower conditional entropy and training loss. This observation aligns with our hypothesis in Section 3. Conversely, in Reverse X scenarios, the R2L model outperforms the L2R(m,n) model. The training and test performance gaps are minimal.

Interestingly, R2L(m) achieves better accuracy than R2L(m,n) in Forward X as conditional entropy decreases. Similarly, L2R(m,n) surpasses L2R(n) in Reverse X. This suggests that **when maintaining the same thinking direction – where computability should remain equivalent – performance improvements can be achieved** by configuring s_i to have lower conditional entropy. This hints that the R2L performance on MCQs can potentially be further improved by configuring the input to predict fewer tokens in the question q, so that the minimum conditional entropy is obtained. We leave this for future exploration.

On the other hand, comparing L2R with R2L(m), where theoretical conditional entropy equals 0, L2R maintains superiority, indicating that **computability likely remains as a key factor**. For the Reverse X task, the accuracy gap between R2L and L2R(n) is smaller than the accuracy gap between L2R(m,n) and L2R(n), suggesting that the conditional entropy may explain more of the performance gap than the computability.

Notably, models achieve higher accuracies on Reverse X compared to their Forward X counterparts, despite similar training loss and conditional entropy values. This disparity could probably be attributed to the closer proximity of choices in Forward X, which inherently increases task difficulty. We provide additional discussion and analysis comparing Forward X and Reverse X in Appendix F.

5 Related Work

Reversal Curse Berglund et al. (2023) first investigates the "reversal curse" in LLMs, which refers to the phenomenon where models trained on forward text data struggle to perform well on inverse search tasks. Allen-Zhu & Li (2023a) further discusses this issue and proposes that augmentation during the pretraining stage can help bridge the knowledge extraction performance gap in reverse entity mapping. In a similar vein, Golovneva et al. (2024) suggest training a unified model that combines text data with augmented reversed or partially reversed data can mitigate the reversal curse. These studies imply that autoregressively-trained language models tend to have a linear and unidirectional thinking process, and certain types of augmentation can facilitate the model in making complex connections between pieces of learned information to enable more intricate cross-referencing. Our research also demonstrates that the autoregressive nature of LLMs may introduce inductive biases rooted from the pretraining corpus. Instead of focusing on the "reversal curse," we suggest that knowledge extraction and reasoning may be more straightforward in the direction with lower conditional entropy.

Order of Reasoning Previous work has also explored the reasoning order's impact on reasoning performance. Vinyals et al. (2015) first demonstrates that the sequence in which input and output data are organized significantly impacts the performance of sequence-to-sequence models and propose to search over possible orders during training to manage unstructured output sets. Recently, Papadopoulos et al. (2024) reveals a surprisingly consistent lower log-perplexity when predicting in L2R versus R2L, despite theoretical expectations of symmetry. The authors attribute this asymmetry to factors like sparsity and computational complexity. We also observe this difference yet we have another hypothesis rationale beyond theirs. Zhang-Li et al. (2024) shows that by reversing the digit order, prioritizing the least significant digit can improve LLMs's performance on arithmetic, which aligns with our findings in section 4.

Previous studies on sequence modeling have also delved into relaxing the conventional “left-to-right” autoregressive dependencies, primarily to facilitate parallel generation (Gu et al., 2018; Ghazvininejad et al., 2019; Gu & Kong, 2021; Zhang et al., 2020) and non-monotonic generation (Welleck et al., 2019; Gu et al., 2019). Text diffusion has recently emerged as a promising approach in terms of planning and controllability (Li et al., 2022; Zhang et al., 2023; Gong et al., 2024). It has shown to be more effective than language models (LLMs), particularly for tasks that require bidirectional reasoning strategies such as sudoku and countdown games (Ye et al., 2024). Alternatively, the Belief State Transformer (BST) (Hu et al., 2025) enhances sequence modeling by using both prefix and suffix inputs to predict subsequent and preceding tokens, effectively capturing a compact belief state for improved goal-conditioned decoding and test-time inference. In contrast, our work primarily investigates the optimal reasoning order for non-generative tasks requiring structured inference.

Multiple-Choice Questions (MCQs) for LLM evaluation MCQs have been widely used for evaluating LLM's reasoning and knowledge extraction abilities. Zheng et al. (2023) demonstrates that LLMs exhibit a selection bias in MCQs, favoring certain option positions, and introduces a debiasing method to mitigate this issue. Pezeshkpour & Hruschka (2023) examines how LLMs' performance on MCQs is influenced by the order of answer options, finding that reordering can lead to huge performance variations. Ghosal et al. (2022) proposes reframing MCQs as a series of binary classifications, demonstrating that this approach significantly improves performance across various models and datasets. Li et al. (2024b) highlights issues like positional biases and discrepancies compared to long-form generated responses, when using MCQs in evaluating LLMs. Wiegreffe et al. (2024) discovers that the prediction of specific answer symbols is primarily attributed to a single middle layer's multi-head self-attention mechanism, with subsequent layers increasing the probability of the chosen answer in the model's vocabulary space. In contrast to the previous work, our work first shows the connection between the preferred reasoning direction and the direction that has lower conditional entropy in MCQ evaluations.

6 Limitations

Task Scope and Generalizability While our findings demonstrate the benefits of R2L factorization in some MCQs, the applicability to other NLP tasks remains unexplored. Our focus on MCQs was intentional to provide a controlled setting for studying factorization effects, as evaluation on generative tasks is not as straightforward. While exploring similar observations in other generative tasks is a valuable avenue for future research, in this paper we limit our discussion to MCQs as a lens to investigate the machinery of LLMs. Additionally, our experiments were limited to models in the 2B-8B parameter range. Due to resource constraints, we were not able to pretrain a SOTA-scale LLM from scratch; future work should explore whether our findings generalize to SOTA models.

Theoretical Framework We propose three hypotheses (calibration, computability, and conditional entropy) to explain different factorization directions' effectiveness, but our analysis in natural language datasets remains qualitative as it is challenging to measure these factors in real language. We used synthetic arithmetic tasks to better understand factors contributing to the L2R and R2L performance gap. Future work should develop rigorous quantitative measures and establish formal theoretical connections between these factors through more sophisticated controlled experiments.

Experimental and Model Design Our evaluation methodology, particularly choosing between $p(q|a_i)$ and $p(a_i|q)$ paradigms, may introduce confounding factors that make isolating pure factorization effects challenging. Future work should develop more robust frameworks that better disentangle these effects from task reformulation. While we focus on standalone L2R and R2L models, incorporating both factorizations through a unified model may further improve performance.

7 Conclusion

In this work, we investigated what makes the preferred thinking direction for LLMs in multiple-choice questions. Through extensive experimentation with models of varying sizes and training datasets, we discovered the surprising finding that R2L factorization can outperform traditional L2R approaches in specific MCQ tasks (4 out of 11 evaluated benchmarks). Our analysis revealed that the effectiveness of each factorization direction may be intrinsically linked to several factors including calibration, computability, and conditional entropy of the downstream task distribution, with lower conditional entropy yielding better performance. We disentangled and validated these factors through controlled simulation studies using arithmetic tasks.

The core contribution of this work lies in bringing this phenomenon to the community's attention and initiating analysis into the underlying factors that might explain why a particular thinking direction is preferred in specific task domains. These findings may suggest the potential for future language model development by revealing the knowledge extraction and reasoning machinery of LLMs and suggesting that alternative factorizations deserve serious consideration in model design. Future work could explore additional factorization strategies beyond L2R and R2L, investigate applications to other types of language tasks, and develop more sophisticated methods for combining different factorizations.

Acknowledgment

We thank Iman Mirzadeh, Mehrdad Farajtabar, Stephen Pulman, Tatiana Likhomanenko, Dan Busbridge, Jason Ramapuram, Huangjie Zheng, Shansan Gong, Shuangfei Zhai, Jiarui Lu and Jiasen Lu for their invaluable insights and discussions that have greatly benefited our project.

References

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, Colin Sandon, and Omid Saremi. How far can transformers reason? the locality barrier and inductive scratchpad. *arXiv preprint arXiv:2406.06467*, 2024.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*, 2023a.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*, 2023b.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL https://aclanthology.org/N19-1245.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NeurIPS*, volume 28, 2015.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *arXiv preprint arXiv:1911.11641*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. In *AAAI*, 2018.

Abhishek Dubey, Ayush Jauhri, Ankit Pandey, Abhishek Kadian, Ahmad Al-Dahle, Alexander Letman, Anshul Mathur, Alan Schelten, Angela Fan Yang, Ankush Goyal, Adam Hartshorn, Ailin Yang, Anirban Mitra, Akhila Sravankumar, Andrey Korenev, Alex Hinsvark, Ananth Rao, Aojun Zhang, Armando Rodriguez, Austin Gregerson, Bogdan Spataru, Baptiste Roziere, Benjamin Biron, Brian Tang, Brian Chern, Caleb Caucheteux, Chitwan Nayak, Chunting Bi, Carlo Marra, Chris McConnell, Christopher Keller, Clement Touret, Chao Wu, Curtis Wong, Carlos Ferrer, Christos Nikolaidis, Daan Allonsius, Da Song, Daniel Pintz, Denis Livshits, David Esiobu, Divyam Choudhary, Dhruv Mahajan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1633. URL https://aclanthology.org/D19-1633.

Deepanway Ghosal, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. Two is better than many? binary classification as an effective approach to multi-choice question answering. *arXiv preprint arXiv:2210.16495*, 2022.

Olga Golovneva, Zeyuan Allen-Zhu, Jason Weston, and Sainbayar Sukhbaatar. Reverse training to nurse the reversal curse. *arXiv preprint arXiv:2403.13799*, 2024.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

Jiatao Gu and Xiang Kong. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 120–133, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.11. URL https://aclanthology.org/2021.findings-acl.11.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *ICLR*, 2018.

Jiatao Gu, Qi Liu, and Kyunghyun Cho. Insertion-based decoding with automatically inferred generation order. *Transactions of the Association for Computational Linguistics*, 7:661–676, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2021.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.564. URL https://aclanthology.org/2021.emnlp-main.564/.

Edward S. Hu, Kwangjun Ahn, Qinghua Liu, Haoran Xu, Manan Tomar, Ada Langford, Dinesh Jayaraman, Alex Lamb, and John Langford. The belief state transformer. *arXiv preprint arXiv:2410.23506*, 2025.

Subbarao Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1):15–18, 2024.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Kumar Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F. Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models, 2024a. https://arxiv.org/abs/2406.11794.

Wangyue Li, Liangzhi Li, Tong Xiang, Xiao Liu, Wei Deng, and Noa Garcia. Can multiple-choice questions really be useful in detecting the abilities of llms? *arXiv preprint arXiv:2403.17752*, 2024b.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-lm improves controllable text generation. *ArXiv*, abs/2205.14217, 2022.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2021.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*, 2024.

Vassilis Papadopoulos, Jérémie Wenger, and Clément Hongler. Arrows of time for large language models. *arXiv preprint arXiv:2401.17505*, 2024.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. SocialIQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

PyTorch Team. torchtune: Pytorch native post-training library. https://github.com/pytorch/torchtune, 2024. Accessed: 2025-02-15.

Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. Llms still can't plan; can lrms? a preliminary evaluation of openai's o1 on planbench. *arXiv preprint arXiv:2409.13373*, 2024.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Sean Welleck, Kianté Brantley, Hal Daumé Iii, and Kyunghyun Cho. Non-monotonic sequential text generation. In *International Conference on Machine Learning*, pp. 6716–6726. PMLR, 2019.

Sarah Wiegreffe, Oyvind Tafjord, Yonatan Belinkov, Hannaneh Hajishirzi, and Ashish Sabharwal. Answer, assemble, ace: Understanding how transformers answer multiple choice questions. *arXiv preprint arXiv:2407.15018*, 2024.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *ACL*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. POINTER: Constrained progressive text generation via insertion-based generative pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8649–8670, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.698. URL https://aclanthology.org/2020.emnlp-main.698.

Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Josh Susskind, and Navdeep Jaitly. Planner: Generating diversified paragraph via latent language diffusion model. In *NeurIPS*, 2023.

Daniel Zhang-Li, Nianyi Lin, Jifan Yu, Zheyuan Zhang, Zijun Yao, Xiaokang Zhang, Lei Hou, Jing Zhang, and Juanzi Li. Reverse that number! decoding order matters in arithmetic learning. *arXiv preprint arXiv:2403.05845*, 2024.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*, 2023.

Appendix

A Model Architecture and Training Hyperparameters

We provide the model architecture of our 2B and 8B models in Table 3. We use no attention dropout and the Normalization ϵ is set to 1×10^{-5}. We employ cosine scheduling for the learning rate, with the number of warmup steps set at 2% of the total iterations. The models are trained on 64 H100 Nvidia GPUs for 60 hours and 208 hours for 2B and 8B models, respectively.

We used the Llama3 tokenizer (Dubey et al., 2024). Despite being trained from a L2R model, Papadopoulos et al. (2024) demonstrated through an ablation study that the direction of tokenizer being trained has minimal effect on the R2L training. Consequently, we adhere to the same reasoning and opt not to train on a R2L tokenizer, as obtaining the necessary training data for Llama3's tokenizer is practically infeasible for fair comparison. Our code implementation is based on Torchtune (Team, 2024), which is released under the BSD 3-Clause License, a permissive open-source license that allows for broad usage and distribution with minimal restrictions.

Table 3: Model architecture specifications for 2B and 8B models.

Parameter	1.5B Model	2B Model	4B Model	8B Model
Vocabulary Size	128,256	128,256	128,256	128,256
Number of Layers	18	24	24	32
Number of Heads	16	16	16	32
Number of KV Heads	16	8	8	8
Embedding Dimension	2048	2,048	3072	4,096
Max Sequence Length	2,048	2048	2,048	8,192
Intermediate Dimension	5,632	7,168	11,264	14,336
RoPE Base	500,000	500,000	500,000	500,000

B Results with Different Model Sizes

We conducted additional experiments to explore performance trends across a broader range of model scales, training 1B and 4B parameter models (R2L and L2R) to complement our existing 2B and 8B models. The full results, presented in Table 4, consistently reinforce our main claim.

Table 4: Results with different model sizes.

	EDU-1.5B		EDU-2B		EDU-4B		EDU-8B	
	L2R	R2L	L2R	R2L	L2R	R2L	L2R	R2L
LogiQA	28.10	**32.65**	27.96	**31.49**	26.58	**32.72**	29.95	**31.03**
OpenbookqQA	38.00	**41.20**	42.40	**44.40**	43.00	**44.60**	45.00	**48.40**
TruthfulQA	23.01	**28.86**	24.36	**28.76**	22.40	**29.00**	24.97	**31.70**
CommensenseQA	41.44	**44.22**	42.92	**45.13**	**50.70**	43.40	39.15	**44.96**
Social_IQA	41.35	**41.65**	**42.78**	42.22	**42.99**	40.00	**44.58**	43.50
AI2_arc	**58.63**	51.8	**60.65**	52.31	**63.14**	53.95	**68.29**	56.22
HellaSwag	**56.46**	43.38	**60.57**	44.34	**65.93**	46.38	**71.60**	49.22
MathQA	**25.52**	24.05	**26.80**	24.86	**27.33**	24.08	**28.77**	25.33
MMLU	33.30	**33.40**	**34.57**	34.35	**36.70**	35.16	**38.90**	37.11
PIQA	**73.23**	59.19	**74.48**	57.13	**76.00**	58.76	**77.80**	59.14
Winogrande	**58.56**	51.77	**60.93**	54.85	**61.80**	54.50	**65.75**	54.70

Specifically, we observe a clear scaling trend: for the tasks where reverse thinking excels (LogiQA, OpenbookQA, TruthfulQA, and Commonsense QA), the performance advantage of R2L over L2R grows as the model size increases. Although we did not scale beyond 8B parameters due to engineering constraints, these extended results demonstrate the robustness of our findings. This trend strongly suggests that the impact of alternative factorizations is unlikely to diminish with even larger models; on the contrary, their benefits may become more significant.

C Comparing Three Paradigms for Reverse Thinking

We present a comparison of the three paradigms discussed in Section 2.1 across all tasks, as shown in Table 5. Paradigm 3 showcases the highest overall performance.

Table 5: Downstream evaluation of three variants to compute s_i.

	Paradigm 1	Paradigm 2	Paradigm 3
LogiQA	31.49%	27.34%	**31.49%**
OpenbookqQA	37.00%	24.60%	**44.40%**
TruthfulQA	24.36%	17.38%	**28.76%**
CommensenseQA	41.85%	27.19%	**45.13%**
Social_IQA	**43.09%**	37.36%	42.22%
AI2_arc	32.19%	45.32%	**52.31%**
HellaSwag	33.73%	43.38%	**44.34%**
MathQA	23.18%	21.07%	**24.86%**
MMLU	33.10%	29.31%	**34.35%**
PIQA	**70.76%**	68.70%	57.13%
Winogrande	54.46%	53.67%	**54.85%**
AVG	38.66%	35.94%	**41.80%**

D Experimental Details on Computing the Conditional Entropy in MCQs

We measure conditional entropy across various MCQs tasks by using ancestral sampling to generate sequences conditioned on each task's input. Generation length can be a confounding issue as longer generation will typically receive larger entropy. To ensure consistent and fair comparison, we constrain the generation to exactly 10 tokens in each direction (L2R and R2L). We make the generation non-stopping by removing the end-of-sentence token. The conditional entropy is first calculated for individual examples within each task, then averaged across all examples to obtain the task-level measure.

E Analysis of How Reverse Thinking Reduces Surface Form Competition in MCQ

Consider a multiple-choice question:

Q: **Who barks?**
A: *dog* B: *cat*

We assume a vocabulary of only three words: *dog*, *cat*, and *puppy*. However, the answer choices are limited to *dog* and *cat*.

Forward Thinking: Surface Form Competition In the forward model, suppose the model predicts the probabilities of each token in the vocabulary as in below:

$$P_{L2R}(\text{dog} \mid q) = 0.3, \quad P_{L2R}(\text{puppy} \mid q) = 0.3, \quad P_{L2R}(\text{cat} \mid q) = 0.4.$$

As Holtzman et al. (2021) shown, here, “dog” and “puppy”, although referring to the same entity, compete as surface forms, potentially impacting the selection of “dog” as an answer.

Reverse Thinking: Alleviating the Surface Form Competition In the reverse thinking, the model should still suffer the surface competition in the prior, suppose we have:

$$P_{R2L}(\text{dog}) = P_{R2L}(\text{puppy}) = 0.2, \quad P_{R2L}(\text{cat}) = 0.6.$$

The reverse conditional probability would would resemble this:

$$P_{R2L}(q \mid \text{dog}) = 0.9, \quad P_{R2L}(q \mid \text{puppy}) = 0.9, \quad P_{R2L}(q \mid \text{cat}) = 0.4.$$

In the reverse thinking R2L paradigm 3, we enforce a uniform prior, which achieves the best performance. We hypothesize that this is due to the alleviation of the surface form competition issue. In this example, by enforcing a uniform prior, surface form competition in the prior distribution disappears, ensuring a fair comparison between “dog” and “cat”. Since we only compare $P_{R2L}(q \mid \text{dog})$ and $P_{R2L}(q \mid \text{cat})$, the probability $P_{R2L}(q \mid \text{puppy})$ becomes irrelevant.

F Additional Results for Simulation Study

We think Forward X is generally more challenging because distinguishing incorrect choices from the correct one can be more challenging than in the Reverse X. Unlike Reverse X where all wrong choices are seen in the dataset, approximately 75% of incorrect choices in Forward X are absent as they cannot be factorized into $m \times n$.

We provide some additional results by performing free ancestral sampling generation (without topK or topP, with temperature set to 1) for the R2L(m) in Forward X. The accuracy of exact match for the generation is reported in below. For reference, the L2R free generation accuracy is 90.4%.

1. Regarding Forward X, R2L(m), free generation of m using $\times n = p_{\text{correct}}$, the exact match accuracy for generative output is **82.1%**.
2. For Forward X, R2L(m), free generation of m using $\times n = p_{\text{incorrect}}$, the generative exact match accuracy remains high at **30.2%**, while the accuracy for random guessing is nearly 0.

These results suggest that the R2L model may map unseen $p_{\text{incorrect}}$ to its neighborhood p_{correct} and execute the same computation as if using p_{correct}. Consequently, the model faces a significant challenge in distinguishing alternative incorrect answers from the correct ones, making alternative incorrect answers hard-negatives.

G Statistical Testing Results

We presents detailed statistical comparisons between L2R and R2L reasoning approaches across four tasks where R2L outperforms L2R: LogiQA, OpenBookQA, TruthfulQA, and CommonsenseQA. Metrics include mean accuracy, standard deviation, and the results of a paired t-test comparing the two approaches. All results were computed using bootstrap sampling with five replicates, each resampled at 80% with replacement, providing a robust estimation of performance variance. Notably, R2L consistently outperformed L2R, with statistically significant improvements (p-value < 0.05) observed across all tasks. The largest relative performance gain appeared in LogiQA, where R2L achieved a mean accuracy increase from 27.04% (L2R) to 32.56% (R2L) with an extremely significant p-value. OpenBookQA also exhibited a clear improvement, rising from 42.80% to 44.40%. TruthfulQA and CommonsenseQA saw meaningful gains as well, with p-values of 0.0072 and 0.0471, respectively, indicating reliable improvements beyond random fluctuation. Overall, these results underscore the effectiveness of reverse thinking (R2L) for improving reasoning tasks.

Table 6: Paired t-test of bootstrap samples (5 replicates, 80% resampled with replacement).

Task	L2R Mean	L2R Std	R2L Mean	R2L Std	p-value
LogiQA	27.04	0.00	32.56	0.0045	0.0000
OpenBookQA	42.80	0.00	44.40	0.0000	0.0000
TruthfulQA	25.08	1.16	28.93	0.80	0.0072
CommonsenseQA	42.17	1.34	44.68	1.70	0.0471