Learning to Imagine: Integrating Counterfactual Thinking in Neural Discrete Reasoning

Moxin Li[1], Fuli Feng[2]*, Hanwang Zhang[3], Xiangnan He[2], Fengbin Zhu[1], Tat-Seng Chua[1]

[1]National University of Singapore, [2]University of Science and Technology of China
[3]Nanyang Technological University,
limoxin@u.nus.edu, fulifeng93@gmail.com,
hanwangzhang@ntu.edu.sg, xiangnanhe@gmail.com,
zhfengbin@gmail.com, dcscts@nus.edu.sg

Abstract

Neural discrete reasoning (NDR) has shown remarkable progress in combining deep models with discrete reasoning. However, we find that existing NDR solution suffers from large performance drop on hypothetical questions, *e.g.,* "*what the annualized rate of return would be if the revenue in 2020 was doubled*". The key to hypothetical question answering (HQA) is counterfactual thinking, which is a natural ability of human reasoning but difficult for deep models. In this work, we devise a Learning to Imagine (L2I) module, which can be seamlessly incorporated into NDR models to perform the imagination of unseen counterfactual. In particular, we formulate counterfactual thinking into two steps: 1) identifying the fact to intervene, and 2) deriving the counterfactual from the fact and assumption, which are designed as neural networks. Based on TAT-QA, we construct a very challenging HQA dataset with 8,283 hypothetical questions. We apply the proposed L2I to TAGOP, the state-of-the-art solution on TAT-QA, validating the rationality and effectiveness of our approach.

1 Introduction

Neural discrete reasoning (Dua et al., 2019) is an emerging technique for machine reading comprehension (Rajpurkar et al., 2016) which aims at answering numerical questions from textual (Dua et al., 2019) or hybrid (Zhu et al., 2021) context[1]. NDR combines deep neural network with discrete and symbolic reasoning (*e.g.,* addition, sorting, or counting) (Dua et al., 2019) and enables the comprehension of complex contexts and compositional questions, which is critical for many practical applications such as automatic diagnosis (Wei et al., 2018) and robo-advisor (Fisch et al., 2019). Existing state-of-the-art NDR models implement the numerical reasoning process as neural network modules (Ran et al., 2019; Herzig et al., 2020; Zhu et al., 2021), *e.g.,* a graph neural network for sorting (Ran et al., 2019; Chen et al., 2020a).

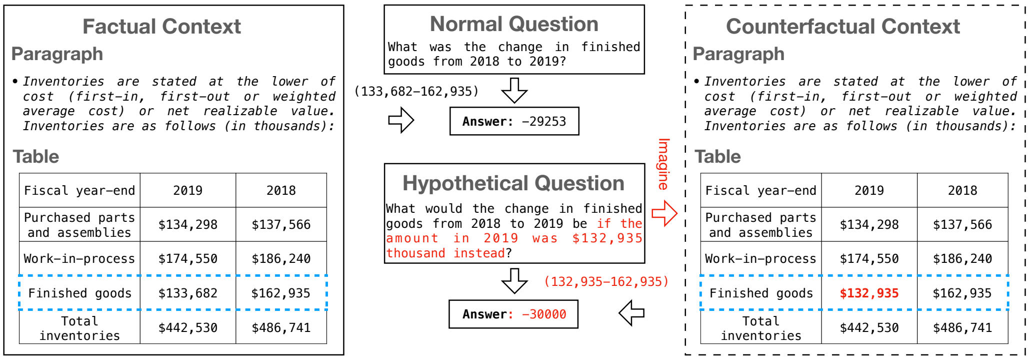
In this work, we extend NDR to *hypothetical question answering* (HQA), where the question consists of an assumption beyond the context (Figure 1). The ability of HQA will undoubtedly enhance the practical use of NDR due to the universality of hypothetical questions. However, current NDR models face severe generalization failure on hypothetical questions. An empirical evidence on such vulnerability is that the state-of-the-art model (Zhu et al., 2021) encounters a sharp performance drop (F1 score drops from 68.6% to 3.8%) on the TAT-QA dataset when changing the questions to be hypothetical by adding a related assumption (see details in Section 2, Table 3). We postulate that the failure is due to unable of imagining the counterfactual context according to the assumption (Figure 1). To pursue such reasoning ability, we resort to the concept of counterfactual thinking (Pearl, 2019) from the theory of causality, which is the ability to imagine and reason over unseen cases based on the seen facts and counterfactual assumptions.

In this light, we consider modeling counterfactual thinking as neural network modules that can be seamlessly incorporated into existing NDR models. One straightforward solution is to model counterfactual thinking as a generation procedure with the fact and assumption as inputs by using a generation model such as GPT (Brown et al., 2020). However, such uncontrollable model (Zou et al., 2021) can hardly generate high-quality context for two reasons: 1) the context is more complex than plain text, which can include a table (Figure 1); and 2) NDR requires a precise context with the correct numbers (Figure 1, *$132,935* for the *finished goods* in *2019*). Therefore, we resort to an alternative approach: constructing the counterfactual

*Corresponding author.

[1] where hybrid includes textual and tabular data in this work

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 57 - 69
May 22-27, 2022 ©2022 Association for Computational Linguistics



Factual Context

Paragraph

- Inventories are stated at the lower of cost (first-in, first-out or weighted average cost) or net realizable value. Inventories are as follows (in thousands):

Table

Fiscal year-end	2019	2018
Purchased parts and assemblies	$134,298	$137,566
Work-in-process	$174,550	$186,240
Finished goods	$133,682	$162,935
Total inventories	$442,530	$486,741

Normal Question

What was the change in finished goods from 2018 to 2019?

Hypothetical Question

What would the change in finished goods from 2018 to 2019 be if the amount in 2019 was $132,935 thousand instead?

Counterfactual Context

Paragraph

- Inventories are stated at the lower of cost (first-in, first-out or weighted average cost) or net realizable value. Inventories are as follows (in thousands):

Table

Fiscal year-end	2019	2018
Purchased parts and assemblies	$134,298	$137,566
Work-in-process	$174,550	$186,240
Finished goods	$132,935	$162,935
Total inventories	$442,530	$486,741

Figure 1: Illustration of hypothetical question and the corresponding counterfactual context to be imagined.

by intervening on the factual context. As shown in Figure 1, the assumption changes one entry in the table, *e.g., $133,682* to *$132,935*. This is coherent with the causal inference theory (Pearl, 2009) where the target variable is intervened according to the hypothetical condition to infer a counterfactual.

We propose Learning to Imagine, where the counterfactual thinking is implemented with two intervening steps: 1) identifying the facts to intervene, and 2) deriving the result of intervention. To pursue accurate context, we derive the intervention with a set of discrete operators such as *SWAP* and *ADD* for imagination. To evaluate the counterfactual thinking ability, we recruit volunteers with domain expertise to construct an HQA dataset based on TAT-QA (Zhu et al., 2021) by posting an assumption for each original question, named TAT-HQA. We apply L2I to TAGOP (Zhu et al., 2021), and obtain a promising solution for HQA. In summary, the main contributions are as follows:

- We highlight the importance of counterfactual thinking in NDR and formulate counterfactual thinking as an intervening procedure to achieve precise imagination.
- We devise the L2I module, which is designed as neural network operations and can be seamlessly incorporated into the NDR model for answering hypothetical questions.
- We construct a challenging HQA dataset and conduct extensive experiments on the dataset, where the performance validates the rationality and effectiveness of the proposed L2I.

2 Hypothetical Question Answering

In the general setting of machine reading comprehension, the task is to answer a question according to the facts in a given context. Formally, it is to learn a function $\boldsymbol{y} = f(\boldsymbol{q}, \boldsymbol{c})$, where $\boldsymbol{y}$, $\boldsymbol{q}$, and $\boldsymbol{c}$ are the word list representing the answer, the question, and the context[2] respectively. This work studies a new and more challenging task that focuses on hypothetical question. As shown in Figure 1, a hypothetical question includes an assumption, *e.g.,* "*if the amount in 2019 was $132,935 thousand instead*". The target of HQA is to learn $\boldsymbol{y} = f(\boldsymbol{q}, \boldsymbol{c}, \boldsymbol{a})$ where $\boldsymbol{a}$ denotes the assumption. The existence of an assumption calls for the imagination of a counterfactual context before inferring the answer, pushing the NDR model to grasp both semantic understanding and counterfactual thinking.

To facilitate the evaluation of HQA and diagnose counterfactual thinking, we construct an HQA dataset based on TAT-QA (Zhu et al., 2021), which is a QA dataset with a mix of tabular and textual context extracted from financial reports. Inspired by previous work on constructing counterfactual samples (Kaushik et al., 2019), we recruit college students with finance-related majors to imagine an intervention based on the factual question and context from TAT-QA which involves numerical thinking, *e.g.,* a change of number. Then they phrase the intervention into an assumption, forming a "*what if*" type of question, and calculate the answer (see an example in Figure 1). To ensure the diversity of the phrasing, annotators are free to generate various phrasing of the assumption, and there is no restriction on the position of the assumption. Usually, the assumption appears either before of after the factual question. Each hypothetical question is related to one factual question from TAT-QA, but each factual question in TAT-QA is not guaranteed to have one hypothetical question. We follow the quality control approaches of annotator training and two-round validation in TAT-QA to guarantee the quality of the hypothetical questions.

[2] Note that recent NDR methods flatten the tabular context (if available) and concatenate it with the textual context. We thus denote the context as a word list for brief notation.

Table 1: Statistics of TAT-HQA dataset by answer type and answer location.

	Tab	Text	Tab-Text	Total
Span	565	16	175	756
Multi-Span	133	1	57	191
Counting	101	5	271	377
Arithmetic	4,423	140	2,396	6,959
Total	5,222	162	2,899	8,283

Table 2: Statistics of TAT-HQA dataset by data split.

Statistics	Train	Val.	Test
# of hybrid contexts	2207	274	277
# of hypothetical questions	6229	823	831
Avg. length of question [words]	23.9	23.6	24.1
Avg. length of assumption [words]	10.58	10.31	10.66

Following TAT-QA, the hypothetical questions are also labeled with four answer types: *arithmetic*, *span*, *count*, and *multi-span*, three types of answer sources: table, text and table-text, and a derivation on how the answer is derived from the context. In total, we obtain 8,283 hypothetical questions, naming it as TAT-HQA. The statistics of TAT-HQA are shown in Table 1. We follow the split of training, testing and validation set of TAT-QA as shown in Table 2.

We conduct a pilot study on the generalization ability of existing NDR models on hypothetical questions. In particular, we evaluate TAGOP (Zhu et al., 2021), which is the state-of-the-art model on TAT-QA (see detailed settings in Section 4.1) by training on TAT-QA and testing on TAT-HQA. In Table 3, the huge performance drop shows that even the state-of-the-art NDR model lacks counterfactual thinking ability.

3 Methodology

We aim to empower NDR models with counterfactual thinking ability. Firstly, we decide to choose the approach of explicitly modeling discrete operations, since existing NDR solutions have demonstrated its superiority (Dua et al., 2019; Ran et al., 2019; Herzig et al., 2020; Zhu et al., 2021). We devise a Learning to Imagine module to model counterfactual thinking (Section 3.1), and then incorporate the L2I module (Section 3.2) into existing NRD methods, followed by a discussion about potential extensions (Section 3.3).

3.1 Learning to Imagine

Functionally speaking, the L2I module aims to construct a counterfactual context based on the factual context and the assumption. We formulate it as: $\boldsymbol{c}' = g(\boldsymbol{c}, \boldsymbol{a})$, where the counterfactual context $\boldsymbol{c}'$ is the status of the context $\boldsymbol{c}$ after the assumption $\boldsymbol{a}$ is executed. Resorting to the language of causality, it can be expressed as the *do*-operation that intervenes a variable to execute the assumption and the *action* to derive the outcome of the intervention[3] (Pearl, 2009). The key to achieving counterfactual thinking in NDR lies in: 1) parsing the assumption to identify the target fact to intervene; and 2) deriving the assumed value to construct the counterfactual context. Taking the hypothetical question in Figure 1 as an example, an ideal L2I should recognize the target variable (*finished goods in 2019*), identify the corresponding fact (*$133,682*), and replace the fact with the assumed value (*$132,935*).

Table 3: Performance of NDR model on TAT-QA and TAT-HQA.

Testing	TAT-QA		TAT-HQA	
	EM	F1	EM	F1
TAGOP	61.3	68.6	2.8	3.8

Two-step Formulation. To this end, we propose a two-step formulation of counterfactual thinking for HQA to perform the identification and derivation. Formally,

$$\textbf{Step 1: } i = r(\boldsymbol{c}, \boldsymbol{a}, \boldsymbol{q}) \qquad (1)$$

$$\textbf{Step 2: } c_i' = d\left(c_i, \boldsymbol{c}, \boldsymbol{a}\right),\ c_j' = \begin{cases} c_j', j = i, \\ c_j, \text{ otherwise.} \end{cases}$$

- **Step 1: Identifying the target fact.** $r(\cdot)$ denotes the tagging function which scans the factual context $\boldsymbol{c}$ to recognize the fact related to the assumption $\boldsymbol{a}$ and the question $\boldsymbol{q}$. i is the word position of the identified fact c_i.
- **Step 2: Deriving intervention result.** $d(\cdot)$ denotes the deriving function that parses the assumption $\boldsymbol{a}$ to infer the discrete operation and the premise to derive the assumed value c_i'. As to the assumption in Figure 1, the derivation requires a *SWAP* operation and a premise *$132,935*. This step then calls for an editing operation to construct the counterfactual context $\boldsymbol{c}'$.

Module Design. Based on the two-step formulation, we then design the L2I module as neural network operations. We have two considerations for the module design: 1) the module should recognize the semantic connection between the assumption and the context, and 2) the module should uniformly support various discrete operations to

[3] Note that we adopt the *do*-expression (Pearl, 2009) of counterfactual.

enable accurate derivation. To this end, we devise four key building blocks for the L2I module:

- *Encoder.* It projects the raw content into latent representation. Inspired by the recent research on NDR, we employ a pre-trained language model (PLM), *i.e.,* RoBERTa (Liu et al., 2019), as the encoder to learn an overall representation of the context, question, and assumption;

$$\boldsymbol{H} = \text{PLM}\left([\text{CLS}, \boldsymbol{c}, \text{SEP}, \{\boldsymbol{q}, \boldsymbol{a}\}, \text{SEP}]\right) \tag{2}$$

  where L and M are the length of the tokenized inputs. CLS and SEP denote the beginning and the separation token of the input. $\{\boldsymbol{q}, \boldsymbol{a}\}$ represents that the relevant position of $\boldsymbol{a}$ to $\boldsymbol{p}$ can vary. We do not assume $\boldsymbol{q}$ to always precede $\boldsymbol{a}$ due to the various location of $\boldsymbol{a}$ in the annotation.

- *Matching block.* It distills the semantic connection between the factual question, the factual context and the hypothetical assumption (Figure 1, "*amount in 2019*" and "*$132,935*"). After applying the token-level self-attention of PLM, we aim to further distill the sequence-level semantic connection between the factual part (the question and the context) and the hypothetical part (the assumption). We obtain the factual and assumption representations by masking $\boldsymbol{H}$ according to the position of the question, the context and the assumption, which splits $\boldsymbol{H}$ into 2 non-overlapping parts. Inspired by the success of cross-attention (Kim et al., 2018) in associating different sources, *e.g.,* image-image (Hou et al., 2019) and image-text (Lu et al., 2019), we adopt cross-attention between the factual representation and the assumption representation, followed by self-attention respectively. Formally, the calculation of the k-th layer is,

$$\begin{aligned}
\boldsymbol{H}_f &= \text{mask}\left(\boldsymbol{H}, \text{pos}(\{c, q\})\right) \\
\boldsymbol{H}_a &= \text{mask}\left(\boldsymbol{H}, \text{pos}(a)\right) \\
\hat{\boldsymbol{H}}_f^k &= \text{MHA}\left(\boldsymbol{H}_f^{k-1}, \boldsymbol{H}_a^{k-1}, \boldsymbol{H}_a^{k-1}\right) \\
\hat{\boldsymbol{H}}_a^k &= \text{MHA}\left(\boldsymbol{H}_a^{k-1}, \boldsymbol{H}_f^{k-1}, \boldsymbol{H}_f^{k-1}\right) \\
\boldsymbol{H}_f^k &= \text{MHA}\left(\hat{\boldsymbol{H}}_f^k, \hat{\boldsymbol{H}}_f^k, \hat{\boldsymbol{H}}_f^k\right) \\
\boldsymbol{H}_a^k &= \text{MHA}\left(\hat{\boldsymbol{H}}_a^k, \hat{\boldsymbol{H}}_a^k, \hat{\boldsymbol{H}}_a^k\right)
\end{aligned}$$

  where MHA$(\cdot)$ denotes the multi-head attention (Vaswani et al., 2017) with a triple of query, key, and value as the input. The residual connection and batch normalization are applied as the default choice. mask$(\cdot)$ denotes the masking operation, and pos(x) is a binary vector with the same length of $\boldsymbol{H}$ denoting the positions of x in the input of PLM.

- *Tagging head.* It models the identification of target fact as a token-wise tagging. Formally,

$$t_i = \begin{cases} 1, \exists(j), argmax(\boldsymbol{p_j}) = 1 \wedge \boldsymbol{h}_j^K \mapsto c_i, \\ 0, \text{ otherwise.} \end{cases} \tag{3}$$
$$\boldsymbol{p_j} = softmax(\text{MLP}\left(\boldsymbol{h}_j^K\right))$$

  where t_i is a binary tag for the fact c_i. c_i will be a target as at least one of its tokens is tagged. We use $\boldsymbol{h}_j^K \mapsto c_i$ to represent the mapping between token and fact, which is true if token j belongs to fact c_i. For each token, we employ a 2-way classifier MLP $\left(\boldsymbol{h}_j^K\right)$ to predict its probability of being tagged as $\boldsymbol{p_j}$ where $argmax(\boldsymbol{p_j}) = 1$ means positive (see Appendix A for more details).

- *Deriving head.* It derives the intervention result for the target fact. To calculate the intervention result, we select a set of commonly used discrete operators such as *SWAP*, *ADD*, and *MINUS* (*cf.* Appendix B). Then, we model the derivation as making a choice across the operators and tagging the premise for executing the operator. In particular, we adopt a tagging head to identify the premise and a multi-way classifier for choosing operators, which is formulated as: $\boldsymbol{o} = softmax(\text{MLP}(\boldsymbol{h}_{CLS}))$. $\boldsymbol{o} \in \mathbb{R}^O$ is a distribution over the operators where O denotes the number of operators. $\boldsymbol{h}_{CLS}$ corresponds to the CLS token in $\boldsymbol{H}$.

3.2 NDR with L2I

Most recent NDR models (Ran et al., 2019; Andor et al., 2019; Chen et al., 2020a; Herzig et al., 2020; Zhu et al., 2021) consist of two main modules: 1) a PLM to encode the context and the question into latent representations, and 2) a reasoning module that chooses the discrete operator and identifies the operands according to the latent representations. As shown in Figure 2, we can seamlessly incorporate the proposed L2I into such NDR model as an intermediate module, which performs imagination before discrete reasoning. In particular, we simply let the reasoning module conduct operand look-up within the counterfactual context constructed by L2I. Besides, we let L2I reuse the PLM in the NDR model to reduce the model complexity and training time.

Model training. Existing NDR methods typically follow the supervised learning paradigm to optimize the model parameters (Dua et al., 2019).

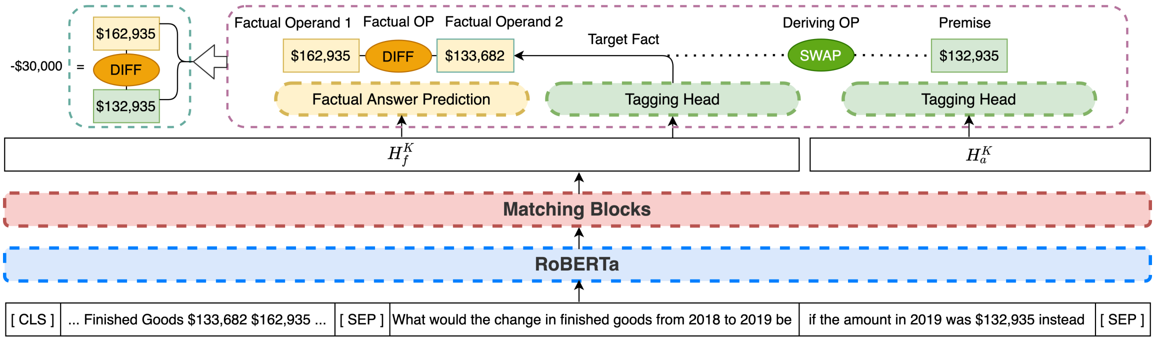


Figure 2: Illustration of NDR equipped with the proposed L2I module for answering hypothetical questions.

Suppose we have a set of labeled questions $\mathcal{D} = \{< \bar{\boldsymbol{y}}, (\boldsymbol{q}, \boldsymbol{c}, \boldsymbol{a}) >\}$, the training objective can be abstracted as $\min_{\boldsymbol{\theta}} \sum_{\mathcal{D}} QA\left(\bar{\boldsymbol{y}}, f(\boldsymbol{q}, \boldsymbol{c}, \boldsymbol{a})\right)$ where $\boldsymbol{\theta}$ denotes model parameters. Note that $QA(\cdot)$ measures the discrepancy between the ground-truth and the predicted answers which can have different formats. For instance, it can be a combination of the cross-entropy (CE) loss over the operand look-up and the CE loss over the choice of discrete operation (Herzig et al., 2020; Yin et al., 2020; Zhu et al., 2021). When applying L2I to an existing NDR method, we keep its question-answering objective unchanged. To optimize the L2I module, we incorporate supervision on the classifiers in the tagging head and deriving head. Formally,

$$\min_{\boldsymbol{\theta}} \sum_{\mathcal{D}} \Big(QA(\bar{\boldsymbol{y}}, f(\boldsymbol{q}, \boldsymbol{c}, \boldsymbol{a})) + \frac{1}{L} \sum_{j<L} \text{CE}\big(\bar{p}_j, \text{MLP}(\boldsymbol{h}_j^K)\big) + \text{CE}\big(\bar{\boldsymbol{o}}, \text{MLP}(\boldsymbol{h}_{CLS})\big) \Big), \tag{4}$$

where $\bar{p}_j \in \{0, 1\}$ denotes the label of the target fact (token j in context) or the premise (token j in assumption); and $\bar{\boldsymbol{o}} \in \mathbb{R}^O$ is the label of the deriving operator (see Appendix C for the details of label construction).

3.3 Discussion

Readers might have raised the following two concerns for L2I: 1) the operators defined are limited, and 2) the operators are tailored to one step of derivation on one target fact. Actually, it is a common approach for current state-of-the-art NDR models to apply a set of defined operators (Ran et al., 2019; Chen et al., 2020a; Zhu et al., 2021). For the first concern, by doing more fine-grained classification on the numerical reasoning process in the dataset, we can derive new operators and simply plug them into L2I. Note that the annotation of numerical intervention of TAT-HQA does not follow the defined operators in Appendix C, but the operators are summarized from the data. Our defined operators can cover over 90% of the training data. For the second concern, we discuss two potential solutions by our L2I framework, and we leave the implementation as future work.

Multi-fact intervention. The assumption $\boldsymbol{a}$ can include intervening multiple facts, *e.g.*, "*if the Finished goods in 2018 and 2019 were both doubled*". Apparently, if the target facts are independent, we can easily handle such an assumption by executing L2I in multiple iterations. In other cases, L2I needs to recognize the relationship among the target facts. If such relationship is available, L2I should be able to handle such cases as the corresponding multi-variable operator is added to the deriving head.

Multi-iteration derivation. In causal inference, a rigorous derivation of an intervention considers the successors of the target variable, *e.g.*, *finished goods in 2019* affects *total inventories in 2019*. Currently, we omit the following iterations in Step 2 of L2I (*cf.* Eq 1). This is because not all successors are necessary for answering the question. For instance, answering the question in Figure 1 does not require the post-intervention value of *total inventories in 2019*. In conventional causal inference, such successors will also be omitted according to the *local surgery* principle (Pearl, 2009). Moreover, we believe that the following iterations can be achieved by the current L2I module in an iterative manner. Assume that NDR model equipped with L2I can answer the hypothetical questions requiring one-iteration derivation (*i.e.*, $c_i \to c_i'$). We can thus derive the value of successors (*e.g.*, $c_i' \to c_j'$) by forming a simple hypothetical question: "*What c_j would be if c_i is c_i'?*" and answering it with the NDR model.

4 Experiments

We conduct experiments on TAT-HQA dataset to answer the following questions: **RQ1:** How does L2I perform on HQA? **RQ2:** What factors influ-

Table 4: Performance of compared methods on the TAT-HQA dataset. The best and the second-best performance *w.r.t.* each metric are highlighted with bold font and underline, respectively. RI means the relative improvement achieved by TAGOP-L2I over the best baseline.

	BERT-RC	NumNet+ V2	TAPAS-WTQ	Hybrider	TAGOP	TAGOP-CLO	TAGOP-L2I	*RI*
EM	4.7±0.4	9.7±0.4	4.7±0.3	4.6±0.2	41.1±0.7	45.4±1.1	**54.4±1.0**	*19.8%*
F_1	10.4±0.5	11.7±0.4	5.9±0.2	4.9±0.1	41.4±0.8	45.7±1.2	**54.7±1.0**	*19.7%*

ence the effectiveness of L2I?

4.1 Experiment Settings

Following Dua et al. (2019) and Zhu et al. (2021), we evaluate the performance with two commonly used metrics: Exact Match (EM) and numerically-focused F_1 score, where higher value (in [0, 100]) means better performance. We tune the hyper-parameters on the validation set, and report the average test performance of five different runs.

Compared methods. To validate the effectiveness of our proposed L2I module, we apply it to TAGOP, obtaining an NDR model for HQA, named TAGOP-L2I. In addition to the vanilla TAGOP, we compare our method against representative methods of traditional QA, numerical QA, tabular QA, and hybrid QA. Besides, we want to select baselines that are effective for learning counterfactual samples. The baselines are: **BERT-RC** (Devlin et al., 2019), a traditional QA method that selects answer spans from the context. **NumNet+ V2** (Ran et al., 2019), a numerical QA method with numerically-aware graph neural network. **TAPAS-WTQ** (Herzig et al., 2020), a tabular QA method that focuses on parsing and understanding tables, pre-trained over tables collected from Wikipedia before training on TAT-HQA. **HyBrider** (Chen et al., 2020c), a hybrid QA method that considers the connection between the table and text. **TAGOP**, a hybrid QA method that performs discrete reasoning over both the tabular and textual contexts. It is the state-of-the-art method on TAT-QA dataset. **TAGOP-CLO**, incorporating the Contrastive Learning Objective (CLO) into the training objective of TAGOP, which is shown to be effective in learning the relationship between factual and counterfactual samples (Liang et al., 2020).

Parameter settings. We implement TAGOP-L2I based on TAGOP[4]. We set the number of cross-attention layers to 3, and fine tune from TAGOP trained on TAT-QA with a learning rate of 5e-5, batch size of 32, and gradient accumulation step of 4. All compared methods are initialized with the model trained on TAT-QA and then fine-tuned on TAT-HQA. For TAGOP-CLO, we conduct max pooling for $\boldsymbol{H}$ and adopt cosine similarity as the distance metric. We select the corresponding factual question as the positive sample and a randomly selected factual question as the negative sample. The weight for the contrastive loss is 0.1.

4.2 Performance Comparison (RQ1)

Overall performance. Table 4 shows the performance of the compared methods on the TAT-HQA dataset. We can observe that: **1)** TAGOP-L2I achieves the best performance among all the compared methods. In particular, it outperforms the best baselines by 19.8% and 19.7% on EM and F_1, respectively. Such significant performance gain validates the effectiveness of the L2I module and reveal the rationality of modeling counterfactual thinking as a neural network module. **2)** TAGOP-CLO outperforms TAGOP by 10.5% and 10.4% on EM and F_1. The only difference between these two methods is that TAGOP-CLO incorporates an extra CLO. The improvement indicates that learning the relationship between the factual and counterfactual samples with CLO provides some clue for counterfactual imagination, yet it is still worse than directly learning to imagine with neural network modules. **3)** As to the remaining methods, their performance has a clear gap between TAGOP, which is consistent with the result on the TAT-QA dataset (Zhu et al., 2021). This is because both datasets have textual and tabular texts, where the ability of TAGOP to perform discrete reasoning across hybrid contexts brings significant advantages. **4)** The performance achieved is still low *w.r.t.* the two metrics (*e.g.,* 54.4→100), showing a large space for future exploration on the challenging TAT-HQA dataset.

Detailed performance. To further investigate the effectiveness of the proposed L2I module, we perform a detailed comparison between TAGOP-L2I and TAGOP *w.r.t.* the discrete operation required in answering the question or counterfactual thinking. We group the questions according to 1) the answer type and 2) the operator to derive the intervention. Table 5 shows the group-wise

[4] https://github.com/NExTplusplus/TAT-QA.

Table 5: Detailed performance of TAGOP-L2I and TAGOP *w.r.t.* answer type and deriving operator type.

Answer Type	TAGOP-L2I		TAGOP	
	EM	F_1	EM	F_1
Span	52.1±3.0	53.4±2.8	46.7±1.7	47.5±1.6
Multi-Span	57.1±3.8	62.0±1.7	51.9±0.0	60.3±0.0
Counting	66.1±4.5	66.1±4.5	52.7±5.6	52.7±5.6
Arithmetic	54.0±1.4	54.0±1.4	39.4±0.8	39.4±0.8

Operator Type	TAGOP-L2I	TAGOP
SWAP	60.5±1.5	47.4±0.9
ADD, MINUS	29.6±1.5	2.0±0.0
MULT, DIV	40.0±14.0	0.0±0.0
PERCENT INC, DEC	56.0±6.8	0.0±0.0
SWAP MIN NUM	5.0±4.1	6.7±8.2

performance. As to answer type (the left half), we have the following observations: **1)** TAGOP-L2I outperforms TAGOP on all groups, showing the superior ability of learning to imagine to all types of questions. **2)** Particularly, on the *arithmetic* group, which is also the largest group (*cf.* Table 1), TAGOP-L2I largely outperforms TAGOP. For this group, the key difference between TAGOP-L2I and TAGOP is whether the derivation of intervention and calculation of the answer are achieved by separate modules. The superior performance of TAGOP-L2I validates the rationality of modeling counterfactual thinking as a separate module. It should be noted that the separation also facilitates the generalization to new operations since the modules can be separately updated. **3)** The performance of TAGOP on *arithmetic* has a large gap with other types, showing that *arithmetic* questions are more difficult to conduct imagination and reasoning even though *arithmetic* makes up the majority of TAT-HQA data. As to TAGOP-L2I, the gap between *arithmetic* question and other types of question largely reduces, validating the effectiveness of learning intervention with discrete operators and neural network modules.

As to operator types (the right half), we observe that: **1)** TAGOP-L2I achieves imagination on the majority of operator types with better performance than TAGOP, yet TAGOP can only achieve imagination on a few operator types. The better performance of TAGOP-L2I is attributed to modeling the deriving operations as specific operators. We thus believe that TAGOP-L2I can generalize well to more deriving operations by simply incorporating the operators, as long as the corresponding training questions are not rare. This result thus reflects the advantage of the unified operator framework adopted by the L2I module, which is consistent with previous work (Andor et al., 2019). **2)** Across the groups, TAGOP achieves relatively good performance on the *SWAP* group, which replaces the target fact with a number in the assumption. It corresponds to the simplest imagination since the assumed value (*i.e.,* c_i') is explicitly mentioned in the assumption. Therefore, the result shows that the NDR model can achieve simple counterfactual thinking by learning to answer hypothetical questions. However, such indirect guidance on imagination fails on the groups requiring more complex imagination, *e.g.,* requiring add or minus. **3)** TAGOP-L2I achieves the worst performance on *SWAP MIN NUM*, which is merely comparable to TAGOP. We suspect the reason is that the operation of *SWAP MIN NUM* is very close to *SWAP*, which may confuse the deriving head when making classification over the operators. To address this issue, it is worth considering the operator relation in the deriving head in the future.

4.3 In-depth Analysis (RQ2)

Study on L2I module design. We then explore the influence of network architecture on the effectiveness of the L2I module from three perspectives: 1) module depth; 2) configuration of the matching block; and 3) the setting of PLM.

Figure 3(a) shows the validation result of TAGOP-L2I as increasing the matching block from 1 to 4 layers. We can observe that: **1)** Stacking more layers does not always bring performance gain. **2)** In particular, three layers of matching block achieve the best performance on TAGOP-L2I. The result indicates that three layers should be sufficient to capture the semantic connection across the context, question and assumption. This is reasonable since the average length of both assumption and question are only around 10 words (*cf.* Table 2).

As to the architecture of the matching block, we evaluate three variants from the default choice *p-s, self-a* which enables parameter sharing across layers (*i.e., ps*) and applies both cross-MHA on the factual and assumption representations and self-MHA for each of them (*i.e., self-a*). The three variants are: 1) *p-s, w/o self-a*, which removes self-MHA; 2) *w/o p-s, self-a*, which disables parameter sharing; and 3) *w/o p-s, w/o self-a*, which adopts both changes. Figure 3(b) shows the performance of the four versions of TAGOP-L2I with

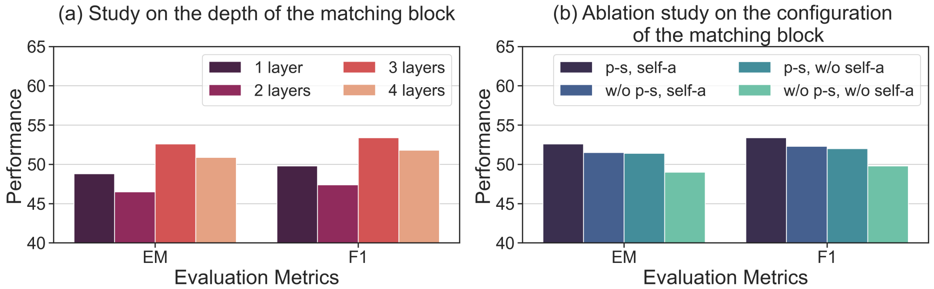


Figure 3: Performance of TAGOP-L2I under difference module configurations.

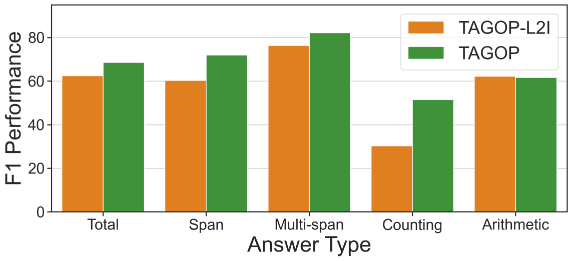


Figure 4: Performance of TAGOP-L2I and the original TAGOP trained on TAT-QA on the test set of TAT-QA.

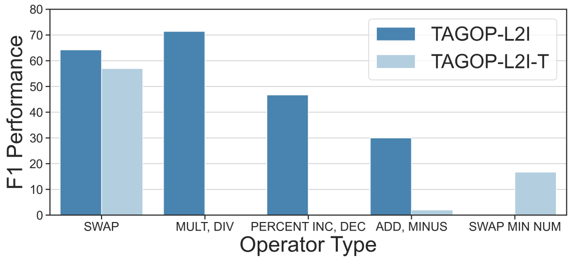


Figure 5: Group-wise performance of TAGOP-L2I and TAGOP-L2I-T *w.r.t.* operator type.

three layers of the matching block. From the figure, we can observe that: **1)** The default choice largely outperforms the variants, validating the rationality of our module design. **2)** Disabling parameter sharing hinders the counterfactual thinking, which indicates that keeping the same parameters through the process of matching factual and assumption representations is beneficial for extracting the semantic correlation. **3)** Removing self-MHA also leads to sharp performance drop, which justifies the contribution of self-MHA in the L2I module. It is thus essential to also separately process the semantic information of the factual and the assumption representations in the matching block.

We also conduct experiments on fixing the parameter of PLM during training on TAT-HQA as initialized by TAT-QA. The performance drops to EM 48.5 and F_1 49.0. Fixing the parameter of PLM largely impedes the performance of TAGOP-L2I on TAT-HQA, showing that encoding factual and hypothetical questions requires different mechanisms. To further investigate the difference in answering factual and hypothetical questions, we test TAGOP-L2I on TAT-QA. The result in Figure 4 shows that training on TAT-HQA causes a performance drop in *counting*, *span* and *multi-span* groups of TAT-QA, and performs similar on the in *arithmetic* group. We conjecture the performance drop in the first three groups is because the question-answering label in TAT-HQA under the same $\boldsymbol{c}$ and $\boldsymbol{q}$ is different from TAT-QA. However, for *arithmetic* questions, the question-answering label for one pair of $\boldsymbol{c}$ and $\boldsymbol{q}$ remains the same between TAT-HQA and TAT-QA, and the intervention is achieved explicitly by deriving operators and tagging head.

Study on L2I training objective. We then investigate the influence of imagination-oriented training objectives on the effectiveness of L2I. In particular, we evaluate a variant TAGOP-L2I-T trained only with the question-answering objective (*i.e.,* $\text{QA}(\cdot)$). That is, TAGOP-L2I-T learns to implicitly imagine the final answer. Figure 5 shows the group-wise performance of TAGOP-L2I and TAGOP-L2I-T *w.r.t.* the type of operator for deriving the intervention. We can observe the followings. **1)** On most groups, TAGOP-L2I largely outperforms TAGOP-L2I-T, demonstrating the rationality of learning to imagine explicitly. **2)** On *SWAP* group TAGOP-L2I-T achieves comparable result to TAGOP-L2I. As *SWAP* is the simplest deriving operator, the result shows that the implicit guidance can achieve simple imagination, yet is still less effective than the explicit manner. **3)** TAGOP-L2I-T achieves better performance on *SWAP MIN NUM* group. As *SWAP MIN NUM* is a rare operator (*cf.* Table 6) and involves the most complex imagination process (*cf.* Appendix B), we conjecture that learning complex operators is more difficult than implicitly learning. This may shed light on the rules of deriving new operators that simple operators with ample training data is preferred over complex operators with less training data.

5 Related Work

Counterfactual thinking. Existing research incorporates counterfactual thinking into deep models from two main perspectives: *counterfactual training* and *counterfactual inference*.

Counterfactual sample has become an emerging data augmentation technique in computer vision (Chen et al., 2020b) and natural language processing (Kaushik et al., 2019) to enhance model robustness. For instance, the technique is applied in visual QA (Chen et al., 2020b; Agrawal et al., 2018; Agarwal et al., 2020; Gokhale et al., 2020), vision-language navigation (Fu et al., 2020; Parvaneh et al., 2020), table entailment (Eisenschlos et al., 2020), sentiment analysis (Kaushik et al., 2019; Yang et al., 2020), natural language inference (Kaushik et al., 2019), named entity recognition (Zeng et al., 2020), and dialogue system (Zhu et al., 2020). Along this line, a series of studies explore how to maximize the effect of counterfactual samples by combining with different learning paradigms, such as adversarial training (Zhu et al., 2020; Fu et al., 2020; Teney et al., 2020), contrastive learning (Liang et al., 2020), causal graph (Gokhale et al., 2020), posterior regularization (Ramakrishnan et al., 2018), and designing new learning paradigms (Gokhale et al., 2020). A few studies along this line also generate counterfactual samples with neural networks (Sauer and Geiger, 2021; Yue et al., 2021). They are inherently different from our work due to their reliance on causal graph and the causal expression of the hypothetical condition for improving robustness. Moreover, they supervise the generation with other related tasks such as image classification. In contrast, we formulate imagination as an explicit learning objective, *i.e.,* learning to imagine. Additionally, in commonsense reasoning, counterfactual samples are also utilized through hyperbole generation (Tian et al., 2021), story generation (Qin et al., 2019) and commonsense QA(Huang et al., 2019), which is also a related yet different strand of research.

Another line of research performs *counterfactual inference* over the predictions of deep model to incorporate counterfactual thinking (Yue et al., 2021; Wang et al., 2021; Niu et al., 2021; Tang et al., 2020). However, they perform counterfactual inference according to causal graph which is not available in NDR tasks.

Neural discrete reasoning. Recent research on NDR focuses on enhancing the discrete reasoning ability of deep models in two main directions: *reasoning with more discrete operations* (Dua et al., 2019; Ran et al., 2019; Chen et al., 2020a) and *reasoning over more complex context*. For instance, NumNet (Ran et al., 2019) and QDGAT (Chen et al., 2020a) leverage graph neural network to enhance comparison oriented operations. GenBERT (Geva et al., 2020) uses pre-trained language models to generate the numerical answer, which breaks the limitation of fixed operators. NMN (Gupta et al., 2019) and FinQA (Chen et al., 2021) model the discrete reasoning process as executing programs. As to extending the context, several studies try to enable the NDR model to operate on context with semi-structured tabular data and hybrid data (Chen et al., 2020c; Herzig et al., 2020; Chen et al., 2021). Our paper studies the hybrid data, yet extends the scope of NDR to hypothetical questions. Moreover, beyond the ability of discrete operations, the main idea is to endow NDR models with the ability to think counterfactually.

6 Conclusion

In this work, we pointed out a key issue of existing NDR models: lacking counterfactual thinking. We proposed an L2I module, which can imagine the counterfactual according to a textual assumption. By applying the proposed module in the NDR model, we enable the model to answer hypothetical questions. We constructed a HQA dataset and conducted extensive experiments on the dataset, which validates the effectiveness of our method.

This work opens up a new research direction about modeling counterfactual thinking through neural network. In the future, we will further extend the L2I from the following perspectives: 1) handling of multiple interventions; 2) rigorous derivation of intervention with consideration of successors; 3) incorporation of the relations across the deriving operators; and 4) construction of complex operators by dynamically combining basic operators. Moreover, we will explore the translation between assumptions in natural language and causal expression to further connect the L2I framework with conventional causal theory, and facilitate automatic causal inference with neural network.

Acknowledgement

This work is supported by Sea-NExT Joint Lab, Singapore MOE AcRF T2, and Natural Science Foundation of China (Grant No. U21B2026).

References

Vedika Agarwal, Rakshith Shetty, and Mario Fritz. 2020. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *CVPR*, pages 9690–9698.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, pages 4971–4980.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving bert a calculator: Finding operations and arguments with reading comprehension. In *EMNLP*, pages 5949–5954.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*, pages 1877–1901.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020a. Question directed graph attention network for numerical reasoning over text. In *EMNLP*, pages 6759–6768.

Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. 2020b. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*, pages 10800–10809.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020c. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *EMNLP*, pages 1026–1036.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, et al. 2021. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*.

Julian Martin Eisenschlos, Syrine Krichine, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. *arXiv preprint arXiv:2010.00571*.

Jill E Fisch, Marion Laboure, and John A Turner. 2019. The emergence of the robo-advisor. *The Disruptive Impact of FinTech on Retirement Systems*, 13.

Tsu-Jui Fu, Xin Eric Wang, Matthew F Peterson, Scott T Grafton, Miguel P Eckstein, and William Yang Wang. 2020. Counterfactual vision-and-language navigation via adversarial path sampler. In *ECCV*, pages 71–86.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *ACL*, pages 946–958.

Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2020. Mutant: A training paradigm for out-of-distribution generalization in visual question answering. *arXiv preprint arXiv:2009.08566*.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019. Neural module networks for reasoning over text. *arXiv preprint arXiv:1912.04971*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *ACL*, pages 4320–4333.

Ruibing Hou, Hong Chang, Bingpeng MA, Shiguang Shan, and Xilin Chen. 2019. Cross attention network for few-shot classification. In *NeurIPS*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *NeurIPS*, pages 1571–1581.

Zujie Liang, Weitao Jiang, Haifeng Hu, and Jiaying Zhu. 2020. Learning to contrast the counterfactual samples for robust visual question answering. In *EMNLP*, pages 3285–3292.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv e-prints*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual vqa: A cause-effect look at language bias. In *CVPR*.

Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Javen Qinfeng Shi, and Anton van den Hengel. 2020. Counterfactual vision-and-language navigation: Unravelling the unseen. In *NeurIPS*, pages 5296–5307.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Judea Pearl. 2019. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. *arXiv preprint arXiv:1909.04076*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. 2018. Overcoming language priors in visual question answering with adversarial regularization. *arXiv preprint arXiv:1810.03649*.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. In *EMNLP*, pages 2474–2484.

Axel Sauer and Andreas Geiger. 2021. Counterfactual generative networks. *ICLR*.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *NeurIPS*, 33.

Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. Learning what makes a difference from counterfactual examples and gradient supervision. *arXiv preprint arXiv:2004.09034*.

Yufei Tian, Nanyun Peng, et al. 2021. Hypogen: Hyperbole generation with commonsense and counterfactual knowledge. *arXiv preprint arXiv:2109.05097*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 6000–6010.

Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. " click" is not equal to" like": Counterfactual recommendation for mitigating clickbait issue. In *SIGIR*.

Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuan-Jing Huang, Kam-Fai Wong, and Xiang Dai. 2018. Task-oriented dialogue system for automatic diagnosis. In *ACL (Volume 2: Short Papers)*, pages 201–207.

Linyi Yang, Eoin Kenny, Tin Lok James Ng, Yi Yang, Barry Smyth, and Ruihai Dong. 2020. Generating plausible counterfactual explanations for deep transformers in financial text classification. In *ICML*, pages 6150–6160.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. In *ACL*, pages 8413–8426.

Zhongqi Yue, Tan Wang, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. 2021. Counterfactual zero-shot and open-set visual recognition. In *CVPR*.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *EMNLP*, pages 7270–7280.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *ACL*.

Qingfu Zhu, Weinan Zhang, Ting Liu, and William Yang Wang. 2020. Counterfactual off-policy training for neural dialogue generation. In *EMNLP*, pages 3438–3448.

Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. Controllable generation from pre-trained language models via inverse prompting. *arXiv preprint arXiv:2103.10685*.

A Working Process of the Tagging Head

The *tagging head* (*cf.* Section 3.1) in L2I identifies the target fact from the factual context, which is formulated as the Inside-Outside (IO) Tagging (Ramshaw and Marcus, 1999). A 2-way classifier, which is a 2-layer MLP followed by softmax, computes the probability of being tagged as negative and positive for each token in the sequence. Then, the positive score for each fact is aggregated by the maximum probability of its tokens. For instance, the fact "$133,682" has four tokens "$", "133", ",", "682" where each token obtains a latent representation from the PLM. The 2-layer MLP takes the latent representation as input to predict the score for each token. The maximum score represents the score of fact "$133,682".

B Deriving Operators

The *deriving head* consists of two steps: 1) selecting the deriving operators; and 2) identifying the premises for the selected operator. The deriving operator is defined as a function $f(T, P)$ over the target fact T and premise P. The value of $f(T, P)$ replaces T in the factual context to form the counterfactual context. In particular, we define eight operators as follows:

- **SWAP**: $f(T, P) = P$.
- **ADD**: $f(T, P) = T + P$.
- **MINUS**: $f(T, P) = T - P$.
- **MULTIPLY**: $f(T, P) = T * P$.
- **DIVISION**: $f(T, P) = T/P$.
- **PERCENT INC**: $f(T, P) = T*(100+P)/100$, where P is a percentage.
- **PERCENT DEC**: $f(T, P) = T * (100 - P)/100$, where P is a percentage.
- **SWAP MIN NUM**: This is a multivariable operator, which intervenes two facts: the target fact T and the sum including the target fact op_2. Apart from swapping T with P, this operator also replaces op_2 with $op_2 - T + P$.

As to the identification of the premise, we simply use the outputs of the tagging head where every fact has a score. We select the fact with the highest score in the context as the target fact and the one in the assumption as the premise.

C Labels for Tagging Head and Deriving Head

Note that each hypothetical question in TAT-HQA corresponds to a question in TAT-QA. Both datasets provide the derivation to answer the question (*e.g., 133,682 - 162,935*), which can be used to construct the ground-truth for training the *tagging head* and *deriving head* of L2I (Equation 4). In particular, we compare the counterfactual derivation with the original derivation. Under the assumption of one-step intervention, we postulate that the counterfactual derivation differs from the original derivation by involving in one more number or substituting one number, where we name the new number in the counterfactual derivation as the premise. By identifying the premise, we construct the label for the tagging head. According to the operator around the premise, we construct the label of the deriving operator. The statistics of the deriving operator are shown in Table 6.

D Accuracy of Deriving Operator Selection and Target Fact Picking

We calculate the accuracy of operator selection and target fact picking of L2I. The average testing result for 5 runs is 96.4% for operator selection, and 82.9% for target fact picking, showing that L2I can select the correct operator and target fact quite precisely. The good performance on selection operators and target facts owes to the superior ability of PLM to understand questions and contexts. We also try a naive lexical match to select operators and target facts. For operator selection, we define a set of keywords(*e.g.,* increase to, decrease by) for the question as a sign of the operator type. For target fact selection, we utilize the word overlap between the assumption and the context to locate the target fact. The accuracy for selecting operators is 89.4%, and for picking up target facts is 52.5%. The gap between L2I and lexical match demonstrates that the generalization ability of PLM plays an important part in operator selection and target fact picking in L2I.

E Computation Resources

We train TAGOP-L2I on a NVIDIA Tesla V100 GPU with 32GB RAM.

	SWAP	ADD	MINUS	MULT	DIV	PERCENT INC	PERCENT DEC	SWAP MIN NUM
Train	4498	274	180	77	7	111	61	52
Val.	540	37	29	9	0	14	3	6
Test	570	32	18	6	1	11	4	12

Table 6: Statistics of the deriving operator.