# Enhancing Self-Consistency and Performance of Pre-Trained Language Models through Natural Language Inference

**Eric Mitchell, Joseph J. Noh, Siyan Li, William S. Armstrong,**
**Ananth Agarwal, Patrick Liu, Chelsea Finn, Christopher D. Manning**
Stanford University
eric.mitchell@cs.stanford.edu

## Abstract

While large pre-trained language models are powerful, their predictions often lack logical consistency across test inputs. For example, a state-of-the-art Macaw question-answering (QA) model answers *Yes* to *Is a sparrow a bird?* and *Does a bird have feet?* but answers *No* to *Does a sparrow have feet?*. To address this failure mode, we propose a framework, Consistency Correction through Relation Detection, or **ConCoRD**, for boosting the consistency and accuracy of pre-trained NLP models using pre-trained natural language inference (NLI) models without fine-tuning or re-training. Given a batch of test inputs, ConCoRD samples several candidate outputs for each input and instantiates a factor graph that accounts for both the model's belief about the likelihood of each answer choice in isolation and the NLI model's beliefs about pair-wise answer choice compatibility. We show that a weighted MaxSAT solver can efficiently compute high-quality answer choices under this factor graph, improving over the raw model's predictions. Our experiments demonstrate that ConCoRD consistently boosts accuracy and consistency of off-the-shelf closed-book QA and VQA models using off-the-shelf NLI models, notably increasing accuracy of LXMERT on ConVQA by 5% absolute. See the project website[1] for code and data.

## 1 Introduction

Reliable and trustworthy AI systems should demonstrate internal *self-consistency*, in the sense that their predictions across inputs should imply logically compatible beliefs about the world. However, even powerful large language models are known to lack self-consistency (Ray et al., 2019; Elazar et al., 2021; Kassner et al., 2021). For example, a question-answering (QA) model that answers the question *Is a sparrow a bird?* and *Does a bird have feet?* with *Yes* is implicitly expressing the belief that *A sparrow is a bird* and *A bird has feet*. If the same model answers the question *Does a sparrow have feet?* with *No*, the model expresses the logically incompatible belief *A sparrow does not have feet*. In such cases, ascertaining the model's "true" belief is difficult, making interpreting and validating its behavior correspondingly challenging.

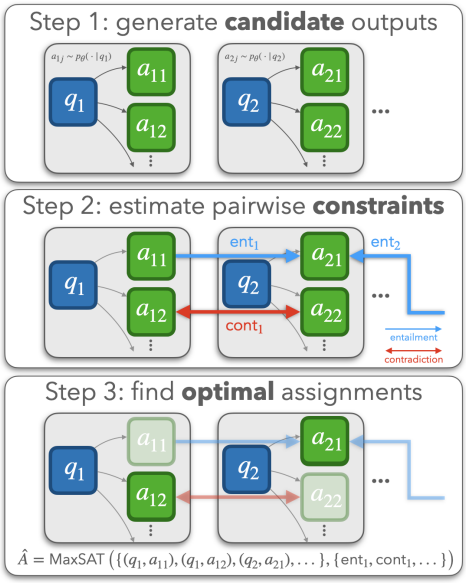



Figure 1: ConCoRD first generates candidate outputs from the base pre-trained model, then estimates soft pairwise constraints between output choices, and finally finds the most satisfactory choices of answers accounting for both the base model and NLI model's beliefs.

Prior work has improved model self-consistency by training with specialized loss functions (Elazar et al., 2021) or data augmentation (Ray et al., 2019), or alternatively re-ranking model predictions based on their mutual self-consistency using pre-written logical constraints, such as "all mammals have fur" (Kassner et al., 2021). However, the first class of methods requires expensive fine-tuning which might be impractical for many practitioners for very large pre-trained models, and re-ranking methods require an explicit collection of the logical relations of interest, making scaling a challenge. Still,

[1] https://ericmitchell.ai/emnlp-2022-concord/



*Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1768
December 7-11, 2022 

re-ranking-based approaches have the benefit of not requiring fine-tuning, and we hypothesize that their scalability limitations may be addressed by *estimating* logical relationships between model predictions on the fly. Specifically, we hypothesize that existing pre-trained natural language inference (NLI) models can estimate logical relationships between an arbitrary pair of model predictions well enough to provide an effective, scalable substitute for explicit collection of such constraints. Leveraging these estimated constraints, we can construct a factor graph representing a probability distribution over model outputs that incorporates both the original model's confidence scores and the NLI model's beliefs about logical relationships.

Our primary contribution is Consistency Correction through Relation Detection, or *ConCoRD*, a framework to improve the consistency and performance of a pre-trained *base language model* without fine-tuning by using more confident and better attested model predictions to override less confident model beliefs. See Figure 1 for an overview. To enable propagation of model beliefs, we estimate pair-wise logical relationships between model predictions using a pre-trained NLI model. Using these pair-wise relationships, we define an undirected graphical model representing a distribution over responses accounting for both the base model's beliefs and the NLI model's estimates of answer compatibility. We efficiently find the approximate mode of this distribution among the base model's top answer choices for each input as the solution of a MaxSAT problem, which consistently produces more accurate and self-consistent predictions than using the raw model predictions. In Section 4.1 we find that ConCoRD produces an 8.1% absolute improvement in F1 of a pre-trained Macaw model (Tafjord and Clark, 2021) on the BeliefBank QA dataset (Kassner et al., 2021). In Section 4.2 we find a 5.0% absolute improvement in accuracy of a pre-trained LXMERT model (Tan and Bansal, 2019) on the ConVQA dataset (Ray et al., 2019), and in Section 4.3 we find that ConCoRD enables test-time *model editing* (Sinitsin et al., 2020; Mitchell et al., 2022), updating model predictions at test time when presented with new information.

## 2 Related Work

Prior work for maintaining consistency in the question-answering space often involves additional training to improve performance. Chen et al. (2021) transform the Natural Questions (Kwiatkowski et al., 2019) dataset question-answer pairs into premise-hypothesis pairs, then uses an NLI model trained on this dataset as a decider for unanswerable questions. Alberti et al. (2019) generate questions from unlabeled texts, then filter them to ensure roundtrip consistency; pre-training on this synthetic set improves performance on SQuAD 2.0 (Rajpurkar et al., 2018) and Natural Questions. Asai and Hajishirzi (2020) augment QA-pairs with their logically symmetric and transitive counterparts through linguistic approaches to enhance cross-dataset QA performance. ConCoRD differs significantly from these question-answering-specific approaches because no fine-tuning of the base model is needed and the methodology is not specific to question-answering.

Similarly to ConCoRD, Kassner et al. (2021) re-rank model predictions by solving an optimization problem defined by a combination of the base model confidence scores and pair-wise constraints representing the logical compatibility of different model predictions stored in a persistent memory, which they call BeliefBank. The key distinguishing property of ConCoRD is the fact that pair-wise constraints between model predictions are dynamically estimated by a pre-trained NLI model, rather than drawn from a fixed, pre-collected set of constraints. Dynamically estimating the constraints has a variety of benefits, eliminating the need for manually collecting the logical constraints of interest, automating the process of determining whether a particular constraint applies to a particular pair of predictions, and likely inheriting improvements in Natural language inference (NLI, MacCartney and Manning (2008)) models over time.

NLI has long been used to maintain logical consistency in generated dialogue utterances (Welleck et al., 2019; Dziri et al., 2019; Song et al., 2020), radiology report domain entities (Miura et al., 2021), and summarization (Laban et al., 2022; Honovich et al., 2022). Perhaps most similarly, Jung et al. (2022) use NLI to estimate constraints between factual statements produced by GPT-3. These prior approaches support our intuition for using NLI models to improve logical consistency among batches of answers. While the authors explore applications of this framework to multi-step reasoning for True/False questions or statements, our work focuses on applying this methodology to more gen-

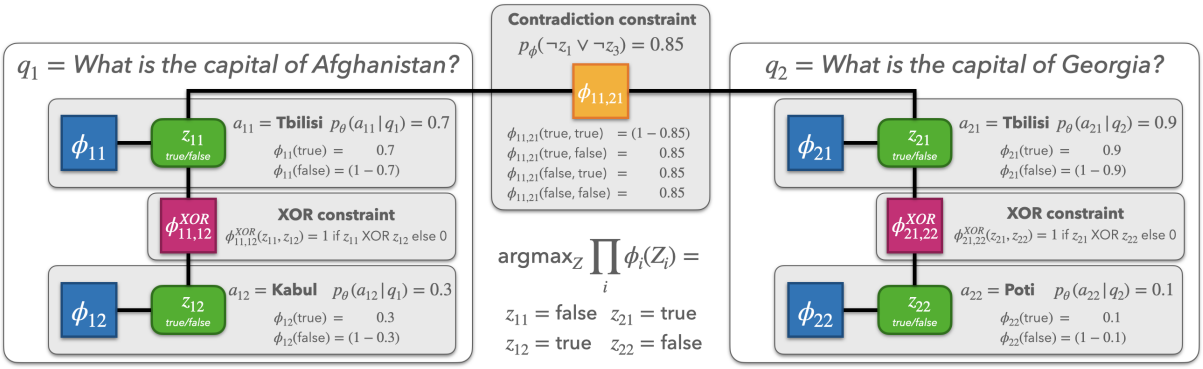



Figure 2: An example factor graph for a simplified batch with two questions, $q_1$ = *What is the capital of Afghanistan?* and $q_2$ = *What is the capital of Georgia?*. Although *Tbilisi* is the most likely answer for both questions, the assignment of variables that is best under the estimated contradiction constraint flips the answer to the first question to *Kabul*. The top-2 answer choices for each question are sampled from the base model, and a soft contradiction constraint is detected between variables $z_1$ (representing the truth of the answer *Tbilisi* for $q_1$) and $z_3$ (representing the truth of the answer *Tbilisi* for $q_2$).

eral settings, such as VQA, open-ended QA, and model editing.

## 3 Consistency Correction through Relation Detection

ConCoRD contains three key components, the *base model*, a *relation model* (typically a pre-trained NLI model), and an *inference procedure* that combines the predictions of the two models into a more accurate and self-consistent set of beliefs. Importantly, both the base model and relation model are pre-trained, off-the-shelf models; ConCoRD does not update any weights or require training data for either model, using only a small validation set for hyperparameter tuning. We next explain the function of each of these components when executing ConCoRD.

### 3.1 Base Model

The core function of the base model in ConCoRD is generating a set of *candidate outputs* for a given input, which are ultimately re-ranked by the inference process (Sec. 3.3). Given a batch of $N$ model queries $Q = \{q_i\}$, the first step of ConCoRD is to generate a set of $J$ candidate outputs for each query $\hat{A}_i = \{\hat{a}_{i1}, ..., \hat{a}_{iJ}\}$, along with their corresponding likelihoods $p_\theta(\hat{a}_{ij}|q_i)$. Note that the candidate outputs need not be an IID sample from the base model; for example, we might use beam search with a diversity bonus to produce a more diverse set of candidates (Vijayakumar et al., 2018). Each pair of query and candidate output forms a *model belief* $b_{ij} = (q_i, \hat{a}_{ij})$; the output of the base model is the complete set of model beliefs $B = \{b_{ij}\}$ and their corresponding *normalized* probabilities $p_\theta^{ij}$[2]. The base models in our experiments are pre-trained question-answering models based on T5-large (Raffel et al., 2020) and pre-trained visual question-answering models such as LXMERT (Tan and Bansal, 2019) and ViLT (Kim et al., 2021).

### 3.2 Relation Model

The relation model $p_\phi(\cdot|x, x')$ estimates the most likely logical relationship between an ordered pair of natural language utterances from the choices {`none`, `fwd-entail`, `contradict`, `equivalence`}.[3] In addition to the model beliefs $B$, we define optional *context statements* $c_{ijk} = C(b_{ij})$, $K$ relevant statements that may be retrieved, generated, or manually written for each model belief. The ability to incorporate context statements enables ConCoRD to modulate model behavior independently for each input in the test batch, rather than reasoning transductively about pairs of test inputs. See Table 3 for examples of model beliefs and context statements. Inputs to the relation model are either pairs of two model beliefs $(b_{ij}, b_{i'j'})$ or pairs of one model belief and one context statement $(b_{ij}, c_{ijk})$. We define the most likely inter-belief relation as $r_{ij,i'j'} = \text{argmax}_r\, p_\phi(r|b_{ij}, b_{i'j'})$, and similarly for belief-context relations $r_{ijk} = \text{argmax}_r\, p_\phi(r|b_{ij}, c_{ijk})$. The output of the relation model is the set of most-likely relations $R = \{r_{ij,i'j'}\} \cup \{r_{ijk}\}$ and their associated probabilities, which we denote as $p_\phi^{ij,i'j'}$ and $p_\phi^{ijk}$. Our experiments use various pre-trained NLI models based on RoBERTa (Liu et al., 2019)

[2]Normalized such that $\sum_j p_\theta^{ij} = 1$.

[3]Because relationships are estimated between ordered pairs of utterances, we can form an equivalence relation if `fwd-entail` is predicted for both orderings of the utterances.

and ALBERT (Lan et al., 2019) as the relation model.

**Question-answer to statement conversion.** While concatenating query $q_i$ and candidate output $\hat{a}_{ij}$ to produce inputs to the relation model is perhaps the simplest approach to estimating soft constraints, we use a statement conversion model to provide inputs to the relation model that are closer to its training distribution. Instead of defining the belief $b_{ij} = (q_i, \hat{a}_{ij})$ as concatenation of $q_i$ and $\hat{a}_{ij}$, we define $b_{ij}$ to be the statement $f_\psi(q_i, \hat{a}_{ij})$, where $f_\psi$ is the conversion model. We fine-tune a small T5 model on a combination of data from (Demszky et al., 2018) and BeliefBank (Kassner et al., 2021) to produce a model that maps a (question, answer) pair into a natural language statement. Details about the fine-tuning procedure and data are provided in Appendix C.

### 3.3 Inference

ConCoRD's inference procedure maps the set of beliefs $B$ and pair-wise relations $R$ into a choice of the most likely belief for each question. To define the inference problem, we first define a binary decision variable $z_{ij}$ representing the estimated truth value of model belief $b_{ij}$. A value of 1 for node $z_{ij}$ in the maximum likelihood configuration means that $\hat{a}_{ij}$ is returned for query $q_i$; the problem includes a constraint that *exactly* one candidate answer is true for each query. The factor graph includes the set of variables $Z = \{z_{ij}\}_{i,j=1,1}^{N,J}$ and various factors (functions mapping a subset of $Z$ to a non-negative scalar) derived from the base model and relation model's beliefs and the hard constraint of returning only one answer per question. Factors are defined such that more desirable configurations of $z_{ij}$ yield a larger *product* of the individual factors. First, unary factors $\phi_{ij}(z_{ij})$ encode the base model's beliefs about the likelihood of specific answers, and are defined as:

$$\phi_{ij}(z_{ij}) = \begin{cases} \frac{p_{ij}}{1-p_{ij}} & \text{if } z_{ij} = 1 \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

where $p_{ij} = p_\theta(\hat{a}_{ij}|q_i)$; in other words, the factor takes the odds ratio if the corresponding statement variable $z_{ij}$ is assigned a truth value of 1; otherwise, the factor takes value 1. In order to encode the hard constraint that exactly one output should be returned for each query, we include a $J$-ary factor $\phi_i(Z_i)$ for each group of nodes $Z_i = \{z_{ij}\}_{j=1}^{J}$, which is equal to 1 for configurations where exactly one of the nodes takes a value of 1, and 0 for all other configurations.

Binary factors $\phi_{ij,i'j'}(z_{ij}, z_{i'j'})$ and optionally $\phi_{ijk}(z_{ij}, c_{ijk})$ encode compatibility between pairs of model beliefs (or model belief-context pairs):

$$\phi_{ij,i'j'}(z_{ij}, z_{i'j'}) = \begin{cases} 1 & \text{if } r_{ij,i'j'}(z_{ij}, z_{i'j'}) \\ 1 - p_\phi^{ij,i'j'} & \text{otherwise} \end{cases}$$

where we define the relation function $r_{ij,i'j'}$ to evaluate to *true* if its arguments satisfy the underlying relation, and *false* otherwise; $\phi_{ijk}(z_{ij}, c_{ijk})$ is defined similarly to $\phi_{ij,i'j'}(z_{ij}, z_{i'j'})$ [4]. The inference problem amounts to finding $\text{argmax}_Z\, \phi(Z)$, where

$$\phi(Z) = \prod_i \phi_i \prod_j \phi_{ij} \left( \prod_{i'j'} \phi_{ij,i'j'} \right) \left( \prod_k \phi_{ijk} \right). \quad (2)$$

An approximate solution to this inference problem can be efficiently found for most problems with a MaxSAT solver such as RC2 (Ignatiev, 2019). We omit arguments to the factors for conciseness. See Figure 2 for a simple example of a factor graph with a single inter-belief constraint and no belief-context constraints.

**Entailment correction.** Consider a belief $b$, a set of its entailed statements $S = \{s_i\}_i$, unary factors $\phi(z_b)$ and $\{\phi(z_{s_i})\}$, and binary factors $P = \{\phi(z_b, z_{s_i})\}_i$. Recall that an entailment relation $r_{bs_i}(z_b, z_{s_i})$ is satisfied (and the binary factor is maximized) if either $z_b = 0$ or *all* $z_{s_i} = 1$. Consequently, as the cardinality of $\{z_{s_i}|z_{s_i} = 0\}$ increases, the more likely it is that $z_b = 0$ will maximize the product of all binary factors $\prod_i \phi(z_b, z_{s_i})$. This is true even if most entailed statements are true, i.e., $|\{z_{s_i}|z_{s_i} = 1\}| \gg |\{z_{s_i}|z_{s_i} = 0\}|$. If most of the statements entailed by a belief are true, assigning the belief to be false due to a small number of (potentially spuriously) false entailed statements may be undesirable. To mitigate this outcome, we experiment with an additional type of factor in which configurations satisfying entailments with both $z_b = 1$ and $z_{s_i} = 1$ are 'rewarded' more than

[4] We use this formulation only to accommodate settings were multiple context statements are retrieved for each query; see Section 4.3. We do not have any $\phi_{ijk}$ factors if we are only using the model's predictions within a batch of test inputs as the premises for reasoning.

other configurations satisfying the entailment:

$$\phi_{b,s_i}(z_b, z_{s_i}) = \begin{cases} 1 & \text{if } z_b, z_{s_i} = 1 \\ 1 - p_\phi^{b,s_i} & \text{if } z_b, z_{s_i} = 0 \\ \sqrt{1 - p_\phi^{b,s_i}} & \text{otherwise} \end{cases}$$

Applying entailment correction consistently improves ConCoRD's performance; see Appendix Table 8 for a dataset-by-dataset breakdown.

### 3.4 Hyperparameters of ConCoRD

We introduce two key hyperparameters to ConCoRD. Because we do not know a priori the relative reliability of the base model and relation model, we introduce the hyperparameter $\beta \in [0, 1]$, corresponding to a trade-off between the predictions of the base model and relation model. A value of $\beta = 1$ corresponds to simply taking the raw predictions of the base model, while $\beta = 0$ corresponds to optimizing purely for answers that are self-consistent according to the relation model, without considering the base model's beliefs. The unary factors in the factor graph become $\phi_{ij}^\beta(z_{ij}) = (\phi_{ij}(z_{ij}))^\beta$ and $\phi_{ij,i'j'}^\beta(z_{ij}, z_{i'j'}) = \left(\phi_{ij,i'j'}(z_{ij}, z_{i'j'})\right)^{1-\beta}$ (and similarly for $\phi_{ijk}^\beta$). In addition to $\beta$, we introduce a threshold $\lambda$ for relation model confidence to filter out low-confidence relation estimates. That is, we discard a relation $r_{ij,i'j'}$ or $r_{ijk}$ if $p_\phi^{ij,i'j'} < \lambda$ or $p_\phi^{ijk} < \lambda$, respectively. In practice, we find that the optimal $\beta$ and $\lambda$ vary across problems, perhaps due to the varying complexity of the model belief and context statements (and therefore the reliability of the relation model's predictions). Therefore, we use the `hyperopt` library (Bergstra et al., 2013) for automated hyperparameter optimization, using the Tree Parzen Estimator (TPE) algorithm to tune $\beta$ and $\lambda$ jointly. We use the optimal hyperparameters found on the validation data for each problem to compute test performance. Appendix H.1 details hyperparameter tuning for each experiment.

## 4 Experiments

Our experiments are broadly designed to answer the high-level question: *can ConCoRD leverage the relational knowledge in pre-trained NLI models to produce more accurate, self-consistent system behavior, without additional data or fine-tuning?* Further, we investigate ConCoRD's applicability to performing test-time *model editing* (Sinitsin et al., 2020; Mitchell et al., 2022), or injection of new information, and ConCoRD's sensitivity to the choice of hyperparameters and types of relations detected.

### 4.1 Internal Consistency in Closed-Book Question-Answering

**Protocol.** To evaluate the accuracy and consistency of a set $B$ of beliefs, Kassner et al. (2021) synthesize a gold standard for those beliefs *and* the inferred relations $R$. Following this prior work, we assume the following is given:

- A set of entities $s_m \in S$
- A set of unary predicates $P_n \in P$
- A collection of "facts" $(P_n(s_m))_i$, whose binary truth value is known
- A directed graph of gold-standard constraints $G(P, E)$, whose edges $(P_n, P_{n'}) \in E$ represent first-order logical formulae $\forall x\,(P_n(x) \to P_{n'}(x))$

From these, we construct simple yes/no questions using natural language templates. For example, for fact $P_n(s_m)$, if entity $s_m$ represents *a lion* and predicate $P_n$ represents *an ability to drink liquids*, the template-generated gold question answer pair $(q_i, a_i)$ is Q: *Is it true that a lion is able to drink liquids?*; A: *Yes.*

These questions are given as input to one of two sizes of a multi-angle question answering model (Tafjord and Clark, 2021), given a multiple choice angle with choices *Yes.* and *No.* The questions and retrieved answers $(q_i, \hat{a}_i)$ form a set of beliefs $B_{s_m}$ for each entity. Since these are closed-book questions, no context statements are supplied; because they are yes/no questions, only one candidate answer is obtained, i.e., $J = 1$. Question-answer to statement conversion is applied to all questions with a default answer of *Yes.* regardless of the answer $\hat{a}_i$, in order to provide the relation model with positive natural language assertions from which to infer sets of relations $R_{s_m}$; where the base model answers $\hat{a}_i$ are *No.* we replace node $z_i$ in the factor graph with its complement. Configurations $Z_{s_m}$ are found for each $s_m \in S$ which maximize Equation 2 given $B_{s_m}, R_{s_m}$ and together form a global solution $Z$.

**Datasets.** Kassner et al. (2021) provide a suitable database with 12,636 facts ("silver facts"), each indicating whether one of 601 predicates relates to one of 85 entities, as well as 4,060 confidence-weighted first-order constraints manually gathered from ConceptNet (Speer et al., 2017), forming a

| | Base | | ConCoRD | | G.C. | |
|---|---|---|---|---|---|---|
| **Model** | F1 | Con. | F1 | Con. | F1 | Con. |
| Mac-Lg | 0.831 | 0.835 | 0.914 | 0.920 | 0.862 | 0.934 |
| Mac-3B | 0.855 | 0.871 | 0.931 | 0.947 | 0.905 | 0.936 |

Table 1: F1 and consistency (1 - $\tau$) for two sizes of Macaw (Tafjord and Clark, 2021) QA models, comparing ConCoRD to a naive QA baseline (Base) and ConCoRD with gold constraints (G.C.). ConCoRD significantly improves both F1 and consistency for both models.

| | Base | | ConCoRD | | Oracle | |
|---|---|---|---|---|---|---|
| **Model** | Acc. | P.C. | Acc. | P.C. | Acc. | P.C. |
| LXM | 0.656 | 0.360 | 0.706 | 0.409 | 0.824 | 0.572 |
| ViLT | 0.784 | 0.489 | 0.804 | 0.548 | 0.882 | 0.690 |

Table 2: ConVQA accuracy (Acc.) and perfect consistency (P.C.) of LXMERT (Tan and Bansal, 2019) and ViLT (Kim et al., 2021) VQA models with and without ConCoRD. ConCoRD significantly improves accuracy and consistency of both models. Oracle performance is top-2 performance, as ConCoRD attempts to select the best of the top 2 answer choices of the base model.

constraint graph $G$. Additionally, they provide 1,072 distinct "calibration facts", each relating one of 7 entities to one of 334 predicates.

We tune $\beta$ and $\lambda$ using a validation set of questions generated from the calibration facts, and evaluate test time performance with questions generated from silver facts.

**Metrics.** We measure accuracy using binary F1 between elements $z_i$ of the configuration Z maximizing $\phi(Z)$ (as in Equation 2), and the truth value of facts $(P_n(s_m))_i$. As in Kassner et al. (2021); we use F1 for evaluation because gold answers are highly biased towards true *No.* answers.

We compute consistency within batches of questions using the complement of of Li et al. (2019)'s conditional constraint violation metric $\tau$, defined here as the proportion of *relevant* gold constraints in $G$ which are *violated*; a constraint $\forall x\,(P_n(x) \rightarrow P_{n'}(x))$ is relevant iff, for some entity $s_m$, there is some belief $b_i \in B_{s_m}$ from fact $(P_n(s_m))_i$ such that $z_i = 1$, *and* there is some belief $b_j \in B_{s_m}$ that corresponds to fact $(P_{n'}(s_m))_j$; the constraint is violated when $z_j = 0$.

**Comparisons.** ConCoRD is evaluated against a naive baseline where only base model answers $\hat{a}_i$ and probabilities are considered. A second baseline (G.C.) performs the inference described in Sec. 3.3, replacing the inferred relations $R$ with the gold constraints from constraint graph $G$, rather than those estimated by the relation model.

**Results.** Results are shown in Table 1. ConCoRD provides an absolute improvement of over 8% in F1 and consistency for Macaw-Large and 7% for Macaw-3B compared to the baseline. Notably, the margin of superiority of the Macaw-3B base model is mostly preserved after applying ConCoRD, suggesting that ConCoRD may provide a significant benefit even for very large models. A surprising result is that ConCoRD shows marked improvements in F1 over the gold constraint baseline, suggesting that the detection and filtering of relations ConCoRD provides may, in this setting, be an improvement over rigid adherence to the logical connections specified *a priori* in Kassner et al. (2021).

### 4.2 Internal Consistency in VQA

**Protocol.** The Visual Question Answering (VQA) task involves a language model generating answers to questions that are directly associated with images. VQA tests for robustness and generalizability of ConCoRD as it introduces an additional layer of difficulty; the task moves away from purely text-based tasks while expanding the answer space to the vocabulary of the LM being used. The questions from the ConVQA dataset (Ray et al., 2019) and its associated images from the Visual Genome dataset (Krishna et al., 2016) provide an apt setting to assess ConCoRD, as the relatedness of questions for each image provide ample opportunity for model self-inconsistency.

The ConVQA dataset consists of a set of images each associated with a group of related questions about the image, such as *What color is the horse?* and *Is the horse brown?* for a picture of a brown horse in a stable. We evaluate ConCoRD with two VQA models, LXMERT (Tan and Bansal, 2019) and ViLT (Kim et al., 2021). For each group of questions $Q_n = \{q_{ni}\}_i$, we sample the top-2 candidate outputs $\{\hat{a}_{ni1}, \hat{a}_{ni2}\}$ for each question, and use a pre-trained NLI model to infer the most likely pair-wise relations $R$ between outputs from different questions. We use the RC2 MaxSAT Solver to estimate the configuration that maximizes Equation 2.

**Metrics.** We report accuracy as the proportion of questions answered correctly across all groups. We infer consistency using a metric previously used in the literature for the ConVQA dataset called "perfect consistency" (Ray et al., 2019). For all groups of related questions, a group is perfectly consistent if all its questions are answered correctly. Perfect consistency then reports the proportion of question groups that were perfectly consistent. While this

| Input & Gold Answer | Generations | Added context |
|---|---|---|
| **Q:** What was the first capital city of Australia? **A:** Melbourne | Canberra; **Melbourne**; Sydney; Inverell | Melbourne was the initial capital following the 1901 Federation of Australia. |
| **Q:** When does the implantation of the embryo occur? **A:** around 9 days after ovulation | 9 to 18 days; **between 6 and 12 days**; after the ovulation; on the 9th week | In humans, implantation of a fertilized ovum is most likely to occur around 9 days after ovulation, however this can range between 6 and 12 days. |

Table 3: Success and failure in editing a model's behavior with ConCoRD by adding new information to the context. The base model's highest confidence answer is Underlined. **Bold** shows ConCoRD's output after inference; with **Teal, bold** showing a successful edit increasing F1 and **Red, bold** showing an edit that reduces F1.

| | F1 | | |
|---|---|---|---|
| **Model** | Base | ConCoRD | Oracle |
| T5-Sm-NQ | 0.207 | 0.225 | 0.281 |
| T5-Lg-NQ | 0.314 | 0.328 | 0.393 |
| T5-3B-NQ | 0.332 | 0.351 | 0.423 |

Table 4: Using ConCoRD to inject contextual information into a model's decisions at test time. Injecting gold Natural Questions contexts consistently improves performance over the base model without requiring fine-tuning.

is not a perfect measure of consistency as it excludes cases in which incorrect answers are consistent with each other, it still serves as a meaningful proxy since the dataset was designed such that any incorrect answer in a question group implies the presence of inconsistency.

**Datasets.** We divide the ConVQA dataset into a "clean" (i.e. human verified and filtered) test set and a non-test set (train + val + test as defined by Ray et al. (2019)). From the non-test set, we sample 10,000 random images equivalent to 123,746 questions to be used as our validation set for tuning our two hyperparameters. We use the clean test set – 725 images and 6,751 questions – to report our final results.

**Comparisons.** ConCoRD is compared with a naive baseline and a top-2 oracle upper bound. The naive baseline is the answer with the highest VQA model probability. Top-2 oracle upper bound selects the correct answer if present within the top-2 predictions of the VQA model. Top-2 is appropriate given our use of the top-2 candidate outputs to generate inferences with NLI models.

**Results.** The final results for ConCoRD, baseline, and oracle upper bound are shown in Table 2. ConCoRD increases the accuracy of LXMERT and ViLT by 5% and 2% respectively, and the consistency of LXMERT and ViLT by 4.9% and 5.9% respectively. Examples in which ConCoRD correctly and incorrectly selects a candidate output different from the baseline output are shown in Figure 4 and Figure 5, respectively. In particular, the incorrect scenarios demonstrate several failure modes that may be in part responsible for the gap between ConCoRD and the oracle upper bound, suggesting further improvements of the components of ConCoRD will also continually improve ConCoRD.

### 4.3 Test-Time Information Injection

**Protocol.** We perform an additional experiment to evaluate ConCoRD's ability to integrate external factual information into its inference process, rather than only using other predictions in the test batch. Such an ability enables editing a model's behavior at test time, without re-training, as new information becomes available. We use the Natural Questions (NQ; Kwiatkowski et al. (2019)) dataset, rather than BeliefBank, to provide more challenging inputs to the relation model. Given a question from NQ, a sentence from the ground truth context document containing information about the answer is retrieved and provided as an additional input to ConCoRD; we constrain the node representing this context variable in the factor graph to be true. Constraints are predicted between each answer choice and the context statement. As in the other experimental settings, hyperparameters are tuned on the validation set and applied on the test set. See Appendix H for tuning procedures.

**Metrics.** Model performance is evaluated using the SQuAD F1 score for overlapping tokens[5], following the same answer normalization protocols, including lower-casing and removing punctuation.

**Datasets.** The NQ development set consists of 7830 open-book question-answer pairs, with both long and short gold annotations in their context passages. Since the NQ test set is not available, we create a test and validation set from the NQ validation questions as follows: we take the first 5000 questions to form our test set, and the rest to be our val set, which we use for hyperparameter tuning. Then each set is filtered such that only

[5] https://worksheets.codalab.org/bundles/0xbcd57bee090b421c982906709c8c27e1

the answerable questions remain. "Answerable" is defined as having a "short answer" span defined in the annotations. This filtering process gives 2713 test entries and 1576 val entries.

**Comparisons.** ConCoRD is compared with a naive baseline and an oracle upper bound. All of these approaches operate on the fixed set of QA model answers for a specific QA model (one of T5-Sm-NQ, T5-Lg-NQ, and T5-3B-NQ), specifically the set of top-4 answers for each question. The naive baseline selects the answer with the highest QA model probability, $\text{argmax}_{\hat{a}_{ij}} p_\theta(\hat{a}_{ij}|q_i)$. The oracle upper bound approach selects the answer that has the best score with the gold short answer span, $\text{argmax}_{\hat{a}_{ij}} F_1(\hat{a}_{ij}, a_{ij})$.

**Results.** The results on the test set using the naive baseline, ConCoRD, and oracle upper-bound are reported in Table 4. ConCoRD always outperforms the naive approach, demonstrating that the framework is useful even when each query input is processed independently (i.e., non-transductively). However, despite providing a relative gain of as high as 8.7% over the naive baseline, there is still a gap between ConCoRD and the oracle. This gap may be attributable to the complexity of the NQ questions and context information compared with the statements in prior experimental settings. Chen et al. (2021) demonstrate a significant gain in calibration performance from training on MultiNLI (Williams et al., 2018) to training on a combination of MultiNLI and their NLI corpus adapted from NQ, perhaps hinting that crucial knowledge present in Natural Questions is not covered in MultiNLI, partially explaining the gap between ConCoRD and oracle F1 performance. Overall, these results suggest that ConCoRD can reason between context statements and model beliefs in addition to pairs of model beliefs, improving performance even with the increased complexity of the data.

**Qualitative Analyses.** Examples of "good" and "bad" edits (edits that improve and decrease the resulting F1-scores respectively) are presented in Table 3, with more in Appendix F. When the correct answer is not available in the candidate outputs, ConCoRD is capable of pushing towards more partially correct answers and those that have more overlap with the context.

### 4.4 Ablating Relation Types

Given that we consider two types of relations in our experiments, contradiction and entailment, it is natural to wonder the relative contribution of these to ConCoRD's performance improvement; Table 5 shows the results of this ablation. We re-run ConCoRD with either entailment or contradiction relations removed, re-tuning the hyperparameters for both of the new settings (contradiction-only or entailment-only). We find that the relative contribution of contradiction and entailment relations varies significantly across models even within the same task, but using both relation types always performs approximately as well or better than using just one, suggesting that both types of detected relations from the NLI model carry useful information. However, we observe in several cases, such as ViLT and the T5 models, that the entailment and contradiction relations may encode somewhat redundant information, as the performance when including either type of constraint alone nearly matches that of using both types.

| Model | Task | F1/Accuracy | | |
|---|---|---|---|---|
| | | ConCoRD | Only cont. | Only ent. |
| Mac-Lg | BB | **0.914** | 0.892 | 0.827 |
| Mac-3B | BB | **0.931** | 0.865 | 0.917 |
| LXM | CVQA | **0.706** | 0.691 | 0.700 |
| ViLT | CVQA | **0.804** | 0.792 | 0.800 |
| T5-Sm-NQ | NQ | 0.225 | 0.225 | 0.225 |
| T5-Lg-NQ | NQ | 0.328 | **0.331** | 0.330 |
| T5-3B-NQ | NQ | **0.351** | 0.349 | 0.350 |

Table 5: Ablating the relation types considered in ConCoRD's inference procedure. The **Only cont.** and **Only ent.** are the results of applying ConCoRD with all entailment or contradiction relations removed, respectively. The **ConCoRD** column is a reproduction of the results from Sections 4.1-4.3, for convenience. Value shown is F1 score for BeliefBank (BB) and Natural Questions (NQ) and accuracy for ConVQA (CVQA). Note that hyperparameters $\beta$ and $\lambda$ are re-tuned on the respective validation set for each setting.

### 4.5 Hyperparameter Sensitivity

We perform several experiments to clarify the relationship between the key hyperparameters, including the specific relation NLI model, $\beta$, and $\lambda$.

**Impact of varying relation model.** Table 6 shows a comparison of ConCoRD's test performance for several NLI models for each setting; notably, the best-performing NLI model is not consistent across problems. While the Albert-XXL model from Nie et al. (2020) is the strongest performing model on NQ, the simpler RoBERTa-Large models outperform it on BeliefBank and ConVQA.

**Sensitivity to $\beta$ and $\lambda$.** Figure 3 shows the performance of ConCoRD on ConVQA with ViLT as

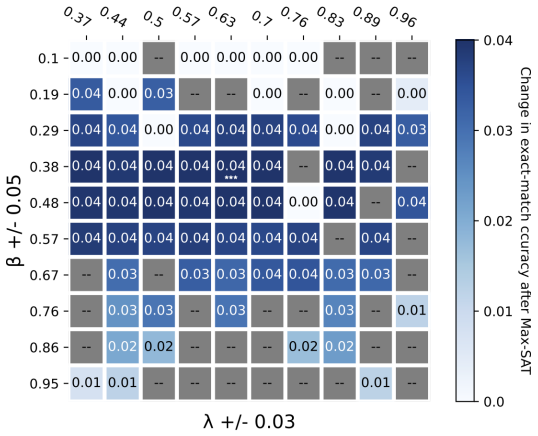



Figure 3: Change in ConCoRD's exact-match validation accuracy as $\lambda$ (the NLI confidence threshold) and $\beta$ (tradeoff between base model and relation model beliefs) vary, holding relation model RoBERTa-Large ANLI constant. By comparing the maximum value within each column or row, we conclude that ConCoRD is relatively robust to the choice of $\lambda$, which the choice of $\beta$ is more important. Values are those encountered during tuning with base model ViLT on ConVQA validation questions. Gray squares correspond to regions not evaluated during search, and asterisks (***) mark the region where the maximum increase in accuracy occurs.

| NLI Model | Data | F1/Accuracy BB | ConVQA | NQ |
|---|---|---|---|---|
| Alb-XXL | ANLI | 0.892 | 0.689 | **0.351** |
| RoB-Lg | ANLI | **0.931** | **0.706** | 0.344 |
| RoB-Lg | MNLI | 0.918 | **0.706** | 0.346 |

Table 6: Comparing ConCoRD's performance for various NLI models on BB (BeliefBank), ConVQA, and NQ. Performance is measured as F1 score between predicted and gold text for BB and NQ, exact match accuracy for ConVQA. We use Macaw 3B for BB results, LXMERT for VQA results and T5-3B for NQ results. The best NLI model(s) in each column are bolded; the best NLI model varies across problems.

$\beta$ (the tradeoff between base model and relation model beliefs) and $\lambda$ (the NLI confidence threshold) are varied, using the values explored during hyperparameter optimization. Section H.2 of the Appendix shows similar visualizations for different VQA experiments. If multiple hyperparameters within a grid element were explored, the best performing configuration is shown. While the maximum value in each column is the same (0.04), indicating that there *exists* a good value of $\beta$ for almost any $\lambda$, the converse is not true; for some values of $\beta$, no good value of $\lambda$ exists. Thus, we conclude that the tradeoff parameter $\beta$ is the more important parameter to tune carefully.

## 5 Discussion & Conclusion

We have presented the ConCoRD framework for enforcing self-consistency in pre-trained language models using relations estimated by pre-trained NLI models, showing that it improves over off-the-shelf performance in a variety of settings without requiring any fine-tuning. Our findings suggest that existing pre-trained NLI models can be a useful building block for boosting performance of NLP systems by providing useful estimates of logical relationships between model predictions across various models and datasets for QA and visual QA.

ConCoRD also suggests several directions for future work. Integrating ConCoRD with methods that *generate* questions likely to elicit useful knowledge for answering the question at hand (Ray et al., 2019; Shwartz et al., 2020) may further improve performance. In addition, integrating a framework such as ConCoRD with recent methods for differentiation through black box combinatorial solvers (Pogančić et al., 2020) may enable training of the entire base model, relation model, and inference pipeline end-to-end, potentially further improving aggregate performance. Finally, ConCoRD's general mechanism of re-ranking predictions by estimating the self-consistency of groups of model predictions is applicable beyond natural language, and future work might investigate its application to problems in vision or sequential decision-making. We hope that ConCoRD may serve as another promising example of integrating both neural and explicit symbolic inference machinery into a broader intelligent system that outperforms any of its components individually.

## 6 Limitations

While our results suggest ConCoRD can effectively leverage additional compute to boost model performance without fine-tuning, our work has some limitations. Although ConCoRD is conceptually applicable to generations from any language model, our work focuses on question-answering settings to leverage existing self-consistency benchmarks. In addition, ConCoRD increases the compute costs of inference, although it does not require fine-tuning. Further, our results suggest that the best NLI model to use for ConCoRD may vary across domains, requiring some tuning. As NLI models improve, we might hope that the final performance of ConCoRD-like systems should also inherit these gains, but Table 6 suggests that the factors that make a particular NLI model well-suited to a particular problem are not obvious, requiring further investigation.

## Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful feedback during the review period, Gabe Mudel, Julie Wang, Cameron Tew, Anthony Tzen, Kevin Yang, and Ian Ng for helpful discussions and assisting with exploratory experiments early on in the project, and Nora Kassner for providing helpful early guidance in configuring the BeliefBank experiments. CF and CM are CIFAR Fellows. EM gratefully acknowledges funding from the Stanford Knight-Hennessy Graduate Fellowship. JN is supported by Stanford University Medical Scientist Training Program grants T32-GM007365 and T32-GM145402. SL acknowledges brownie bites from Target for providing a crucial fuel source for late night experiment-running.

## References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5642–5650, Online. Association for Computational Linguistics.

James Bergstra, Dan Yamins, David D Cox, et al. 2013. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, volume 13, page 20. Citeseer.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. 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)*, pages 3806–3812, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161–175, Dublin, Ireland. Association for Computational Linguistics.

Alexey Ignatiev. 2019. Rc2: an efficient maxsat solver. *J. Satisf. Boolean Model. Comput.*, 11:53–64.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. ViLT: Vision-and-language transformer without convolution or region supervision. In *ICML*, pages 5583–5594.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *International Journal of Computer Vision*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Sori-

cut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. 2019. A logic-driven framework for consistency of neural 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)*, pages 3924–3935, Hong Kong, China. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Hans-Andrea Loeliger. 2008. An introduction to factor graphs. `https://people.binf.ku.dk/~thamelry/MLSB08/hal.pdf`.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK. Coling 2008 Organizing Committee.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. Fast model editing at scale. *ICLR*.

Yasuhide Miura, Yuhao Zhang, Emily Tsai, Curtis Langlotz, and Dan Jurafsky. 2021. Improving factual completeness and consistency of image-to-text radiology report generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5288–5304, Online. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. 2020. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. 2019. Sunny and dark outside?! improving answer consistency in VQA through entailed question generation. 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)*, pages 5860–5865, Hong Kong, China. Association for Computational Linguistics.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online. Association for Computational Linguistics.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. In *ICLR*.

Haoyu Song, Wei-Nan Zhang, Jingwen Hu, and Ting Liu. 2020. Generating persona consistent dialogues by exploiting natural language inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8878–8885.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Oyvind Tafjord and Peter Clark. 2021. General-purpose question-answering with macaw. *CoRR*, abs/2109.02593.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. 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)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

## A Reproducing Macaw-Large Examples

The following configuration reproduces the Macaw-Large behavior noted in the abstract and the introduction at `https://huggingface.co/allenai/macaw-large`.

```
$answer$ ; $question$ = Is a sparrow a bird? ; $mcoptions$ = (A) Yes. (B) No. ;
$answer$ ; $question$ = Does a bird have feet? ; $mcoptions$ = (A) Yes. (B) No. ;
$answer$ ; $question$ = Does a sparrow have feet? ; $mcoptions$ = (A) Yes. (B) No. ;
```

## B Factor Graph Overview

A factor graph is a factorization of a function $f$ mapping a set of $n$ variables $Z = \{z_j\}_{j=1}^{n}$ to a non-negative scalar. The factorization is represented as a bipartite graph containing *variable nodes* and *factors*; each $z_j$ is represented by one variable node, and each factor $\phi_i$ maps a subset of the variable nodes $Z_i$ to a non-negative scalar. The value of the function is computed as $f(Z) = \prod_i(Z_i)$. See Loeliger (2008) for a more complete reference.

## C Question-Answer to Statement Conversion Model Details

To convert question-answer pairs into declarative statements, we combine data from the Question to Declarative Sentence (QA2D) (Demszky et al., 2018) and BeliefBank (Kassner et al., 2021) datasets to fine-tune a T5-base sequence-to-sequence model. QA2D contains question-answer pairs from five QA datasets; 95% of the pairs are from SQuAD (Rajpurkar et al., 2016). The gold statements are from Amazon Mechanical Turk. The BeliefBank questions are created from silver facts using natural language templates as in Section 4.1, and the yes/no answers are from the known binary truth values of these facts. Our training dataset is composed of the full QA2D training dataset of 61k question-answer pairs and half of the BeliefBank silver facts, for a total of 67k training examples. Likewise, the validation dataset consists of the full QA2D validation dataset of 10k pairs and half the BeliefBank silver facts, for a total of 16k validation pairs.

The input to the QA statement conversion model is the concatenation of the question-answer pair $q_i \parallel a_i$. Accuracy is evaluated by comparing the output sequence tokens to the gold sequence tokens. Training occurs with a learning rate of $1e^{-4}$ for a maximum of 50 steps, where each step consists of 32k training examples, with early stopping if validation loss does not decrease for 6 consecutive steps. We ran the fine-tuning on NVIDIA GeForce RTX 3090 GPU. Fine-tuning ended after 14 steps with a final training accuracy of 0.764 and validation accuracy of 0.628. This took approximately 40 minutes. Table 7 demonstrates the model's performance on a few validation examples.

## D Additional Modifications to ConCoRD

A timeout for solvers is imposed in order to prevent the RC2 MaxSAT solver from running optimization indefinitely. The average solve time per question was <4 ms for closed-book QA, <1 ms for VQA and <20 ms for NQ (for NQ, the solve time is < 1/10th of the time needed for a forward pass through the QA and NLI models). We found only one batch of test questions for the closed-book QA task and VQA task where the solver couldn't find a solution efficiently, so we set a short timeout (30s for CBQA, 10s for VQA, none required for NQ).

We also de-duplicate the list of inferred constraints before passing the statement and constraint groups through the MaxSAT solver so that only the highest-weighted constraints would remain among their duplicates.

## E Entailment Correction Ablations

Table 8 shows the effects of entailment correction on ConCoRD test performance in closed-book question answering and VQA experiments for different choices of base model, using the NLI relation model resulting in the best test set performance (RoBERTa-Large-MNLI).

| Dataset | Input | Output | Gold statement |
|---|---|---|---|
| SQuAD | Who established Yale's residential college system? Edward S. Harkness | Edward S. Harkness established Yale's residential college system. | Edward S. Harkness established Yale's residential college system. |
| SQuAD | How did Kuhn view the history of science? competing paradigms or conceptual systems | Kuhn viewed the history of science as **a competing paradigm or conceptual system**. | Kuhn viewed the history of science as **competing paradigms or conceptual systems**. |
| BeliefBank | Is it true that a poodle is a river? No | A poodle is not a river. | A poodle is not a river. |
| BeliefBank | Is a pigeon a living thing? Yes | A pigeon is a living thing. | A pigeon is a living thing. |

Table 7: The QA statement conversion model outputs declarative statements from question-answer pairs. Out of the four validation examples presented, three are correct. The **Red, bolded** portion of the output of the second example indicates how it differs from the **Teal, bolded** corresponding portion of the gold statement.

| | F1/Accuracy | | |
|---|---|---|---|
| Model | Naive | w. E.C. | w/o. E.C. |
| Mac-Lg+Rob/ANLI | 0.831 | **0.914** | 0.909 |
| Mac-3B+Rob/ANLI | 0.855 | **0.931** | 0.886 |
| LXMERT+Rob/MNLI | 0.656 | **0.706** | 0.701 |
| LXMERT+Rob/ANLI | 0.656 | **0.706** | 0.693 |
| ViLT+Rob/MNLI | 0.784 | 0.804 | **0.810** |
| ViLT+Rob/ANLI | 0.784 | **0.814** | 0.807 |

Table 8: Comparison of ConCoRD test performance vs. baseline with and without entailment correction (E.C.) across base+relation models for closed-book question answering (Macaw) and VQA (LXMERT, ViLT) experiments (F1 for closed-book QA, exact-match accuracy for VQA), showing that the entailment correction improves performance for most configurations.

| Experiment | Model | Good Flips | Bad Flips |
|---|---|---|---|
| BeliefBank | Macaw-3B | 723 | 277 |
| VQA | LXMERT | 576 | 238 |
| NQ | T5-3B-NQ | 168 | 69 |

Table 9: The numbers of good and bad flips in each of the experiments performed. We define flips as choosing a different candidate from the naive baseline for the multiple choice experiments, and a binary truth value flip for BeliefBank. "Good" flips are flips that improves performance, and "bad" flips are those that are detrimental to performance.

## F Additional "Good" and "Bad" Edit Pairs

More examples of good and bad edits in the Editing experiment are presented in Table 10. We also include good (Figure 4)and bad flip (Figure 5) examples from the VQA dataset. For the bad flip examples in VQA, we include different failure modes to demonstrate the types of potential ConCoRD errors.

## G Good and Bad Flips

For each set of experiments on the test set, we report the numbers of good and bad flips made by ConCoRD in Table 9. It can be observed that the number of good flips is consistently significantly higher than that of bad flips.

## H Hyperparameter Search Details

### H.1 Experiments

#### H.1.1 Closed-Book Question Answering

Hyperparameters (Section 3.4) are tuned jointly using `hyperopt` on the BeliefBank calibration dataset (Section 4.1). The search space of $\beta$ is uniform between $[0.05, 1.0]$, and for $\lambda$ it is uniform between $[0.5, 1.0]$. `hyperopt` optimizes cumulative F1 across all entity batches for 300 trials. To speed-up tuning, we created caches of model beliefs $B_{s_m}$ and relation sets $R_{s_m}$ for each calibration entity $s_m$. This was run on NVIDIA GeForce RTX 3090 GPU, and the largest NLI models took up to two hours to complete. Using these caches, `hyperopt` tuning completes in less than an hour on CPU. The best performance on the calibration facts for each of the base Macaw models is reported in Table 11. The results show that $\beta$ is higher for the better base model Macaw-3B.

#### H.1.2 VQA

Hyperparameters are tuned jointly using `hyperopt`. The search space for $\beta$ is uniform over $[0.05, 1]$, for $\lambda$ it is uniform over $[\frac{1}{3}, 1]$. A total of 100 trials were performed, updating parameters using TPE, on an AWS `g4dn.xlarge` EC2 instance. Each search took less than one hour. Table 12 shows the selected parameters and their exact-match accuracy on validation questions.

| Model | Input & Gold Answer | Generations | Added context |
|---|---|---|---|
| T5-Sm-NQ | **Q:** Who was the declaration of independence written for?<br>**A:** the Second Continental Congress | Second Continental Congress; **the United States**; the British Crown; Great Britain | The United States Declaration of Independence is the statement adopted by the Second Continental Congress meeting at the Pennsylvania State House ( Independence Hall ) in Philadelphia on July 4 , 1776 , which announced that the thirteen American colonies , then at war with the Kingdom of Great Britain , regarded themselves as thirteen independent sovereign states , no longer under British rule . |
| T5-Sm-NQ | **Q:** What is the scientific name for the calf muscle<br>**A:** gastrocnemius muscle | The serratus calf muscle; **gastrocnemius muscle**; The serratus calfi; The serratus muscle | Along with the soleus muscle , the gastrocnemius forms half of the calf muscle . |
| T5-3B-NQ | **Q:** Who is the actor that plays Dr. Sean Murphy<br>**A:** Freddie Highmore | Freddie Highmore; **Daryl "Chill" Mitchell**; Dylan Christopher Minnette; Javier Muoz | The series stars Freddie Highmore as Shaun Murphy , a young surgical resident with autism and savant syndrome at San Jose St. Bonaventure Hospital ., Freddie Highmore as Shaun Murphy : A surgical resident with autism and savant syndrome . |
| T5-3B-NQ | **Q:** Who is the founder of the Ubuntu project<br>**A:** Mark Richard Shuttleworth | Linus Torvalds; **Mark Shuttleworth**; Richard St. John Hopper; Richard St. John Redmond | Mark Richard Shuttleworth ( born 18 September 1973 ) is a South African entrepreneur who is the founder and CEO of Canonical Ltd. , the company behind the development of the Linux - based Ubuntu operating system . |

Table 10: Editing a model's behavior by adding new information to the context. The Underlined generation is the answer with the highest QA model confidence. The **Bolded** generation is what ConCoRD selects after NLI inference. **Teal, bolded** generations indicate that ConCoRD selects a generation with higher token overlap F1, while **Red, bolded** generations indicate that ConCoRD selects a worse generation.

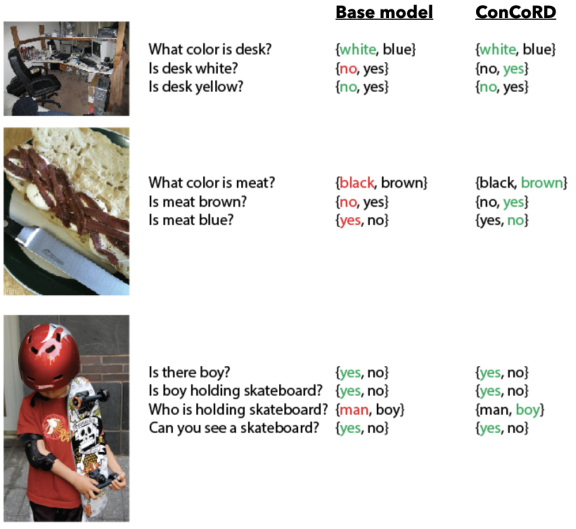



Figure 4: "Good" flip examples from the VQA experiments. The green texts mark the correctly selected answers, while the red texts indicate incorrectly selected answers.

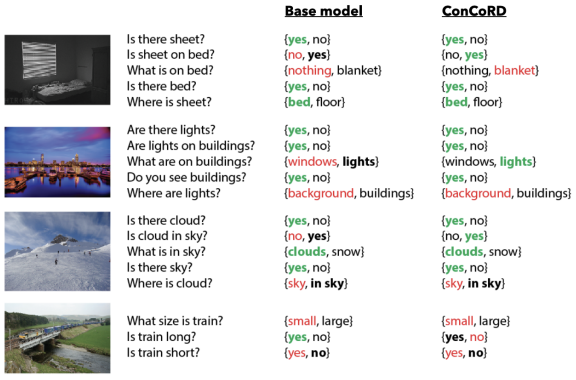



Figure 5: "Bad" flip examples from the VQA experiments. The green texts mark the correctly selected answers, while the red texts indicate the incorrectly selected answers. The **bolded** texts are the correct answers, if generated within the top-2 predictions. From top to bottom, the first image is an example of when the correct answer, "sheet," was not contained in the predicted answers. The second image is an example of when the conversion of QA pair to statement did not occur as intended and the NLI failed to generate the appropriate inferences that could be used to inform correction of "background" to "buildings. The third image shows an example of when an "incorrect" answer (sky) is effectively the same as the "correct" answer (in sky)–only semantically different. The fourth image shows an example of when the model strongly believed in an incorrect answer and changed another correct answer.

| Model | F1 | $\beta$ | $\lambda$ | E.C. |
|---|---|---|---|---|
| Macaw-Large | 0.919 | 0.753 | 0.855 | True |
| Macaw-3B | 0.94 | 0.804 | 0.873 | True |

Table 11: Validation performance on the BeliefBank calibration facts. Both models achieve best validation performance with the RoBERTa-Large ANLI model.

| VQA | Acc. | $\beta$ | $\lambda$ | E.C. |
|---|---|---|---|---|
| LXMERT | 0.691 | 0.208 | 0.805 | True |
| ViLT | 0.787 | 0.395 | 0.772 | True |

Table 12: Validation performance on VQA. Both models achieve best validation performance with the RoBERTa-Large MNLI model.

### H.1.3 Information Injection with Natural Questions

For this round of experiments, we lower the bounds for $\beta$ and $\lambda$ after some initial trials. The bounds of $\beta$ are $[0, 0.5]$ and the bounds of $\lambda$ are $[0, 0.6]$. We run `hyperopt` for 200 trials (often taking approximately 2 to 3 hours on an NVIDIA GeForce RTX 3090 GPU) for each of the three NLI models. `Hyperopt` optimizes for the highest token-overlapping F1 score in this experiment.

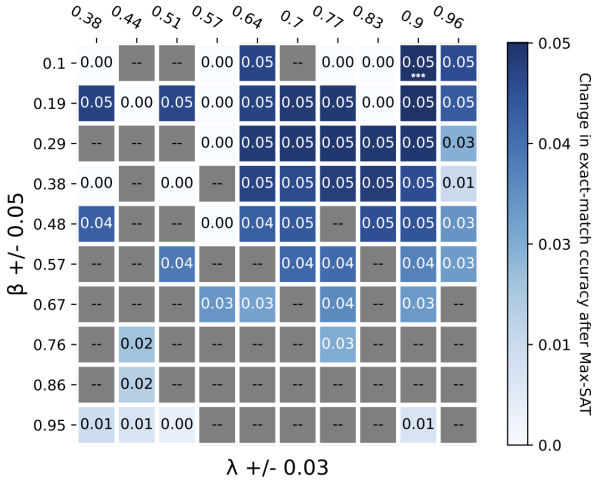
We report the best validation performance of each of the QA base models in Table 13.

| Model | F1 | $\beta$ | $\lambda$ | E.C. |
|---|---|---|---|---|
| T5-Small | 0.227 | 0.112 | 0.540 | True |
| T5-Large | 0.331 | 0.081 | 0.413 | False |
| T5-3B | 0.353 | 0.072 | 0.477 | True |

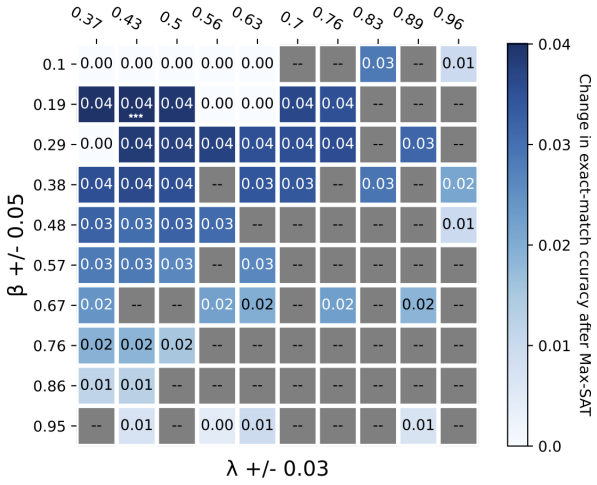
Table 13: Validation performance on NQ. All models achieve best validation performance with the ALBERT ANLI model.

## H.2 Visualizing Hyperparameter Search

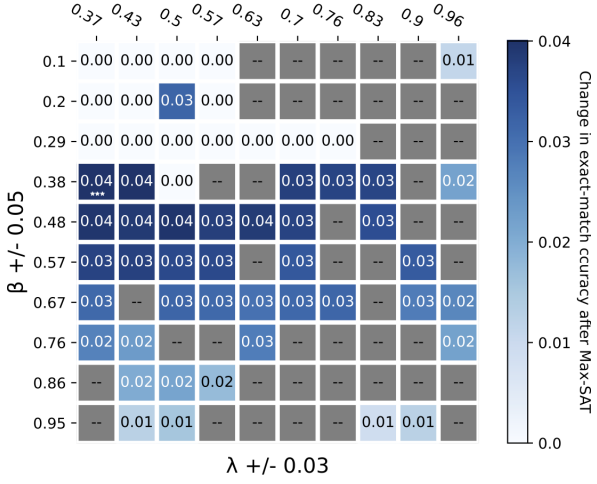
Figure 6 shows increases in exact-match accuracy as they vary with choices of $\lambda$, $\beta$, for additional choices of base model for a VQA task, with and without entailment correction, complementing figure 3. Interestingly, choosing a different base model does noticeably effect the optimum value of $\beta$; between figures 6b and 6c we see the near-optimal region shift towards a value of $\beta$ that gives higher confidence in the base model where the base model produces "better" answers. However, the increase in accuracy is similar, suggesting that with appropriate selection of $\beta$, ConCoRD can offer similar improvements over a range of choices of base model.




(a) Base model LXMERT, with entailment correction.




(b) Base model LXMERT, without entailment correction.




(c) Base model ViLT, without entailment correction.

Figure 6: As in figure 3, we show changes in exact-match validation accuracy as a function of confidence threshold $\lambda$ and tradeoff parameter $\beta$, with several choices of base model, with and without an entailment correction, holding relation model RoBERTa-Large ANLI constant.