# Consistent Joint Decision-Making with Heterogeneous Learning Models

Anonymous ACL submission

## Abstract

This paper introduces a novel decision-making framework that promotes consistency among decisions made by diverse models while utilizing external knowledge. Leveraging the Integer Linear Programming (ILP) framework, we map predictions from various models into globally normalized and comparable values by incorporating information about decisions' prior probability, confidence (uncertainty), and the models' expected accuracy. Our empirical study demonstrates the superiority of our approach over conventional baselines on multiple datasets.

## 1 Introduction

The rapid advance of AI has led to the widespread use of neural networks in tackling complex tasks that involve multiple output decisions, which may be derived from various models (Liu et al., 2022; Wang et al., 2022). However, these decisions are interrelated within the same problem and must conform to specific constraints. For example, to comprehend procedural text, multiple neural models collaborate to establish temporal relationships between actions, reveal semantic relations, and discern entity properties like location and temperature (Faghihi et al., 2023a; Bosselut et al., 2018; Jiang et al., 2023). Each model exhibits distinct decision characteristics, output sizes, uncertainty levels, and varying excepted accuracy levels. Resolving inconsistencies and aligning these diverse neural decisions is crucial for a comprehensive understanding of the underlying process.

In many instances, raw model outputs lack usability without enforcing consistency. In tasks like hierarchical image classification, with independent models for each hierarchy level, outputs should adhere to the known hierarchical relationships. For example, the combination "Plant, Chair, Armchair" lacks validity and requires post-processing for downstream applications. A similar requirement extends to generative models in text summarization (Lu et al., 2021) and image captioning (Anderson et al., 2017). Prior studies have proposed techniques for handling inconsistencies in correlated decisions during both inference (Freitag and Al-Onaizan, 2017; Scholak et al., 2021; Dahlmeier and Ng, 2012; Chang et al., 2012; Guo et al., 2021) and training (Hu et al., 2016; Nandwani et al., 2019; Xu et al., 2018) of neural models. This paper focuses on resolving these inconsistencies at inference, where the goal is to ensure that outputs align with task constraints while preserving or enhancing the original model performance without training.

In addressing decision inconsistencies, Integer Linear Programming (ILP) (Roth and Yih, 2005) stands out as a robust approach. ILP is a global optimization framework that seeks to find the best configuration of variables while meeting specified constraints. It is known for its efficiency and capability to produce globally optimal solutions, distinguishing it from alternatives like beam search. The ILP formulation is as follows:

$$\begin{aligned} \text{Objective : Maximize} \quad & P^\top y \\ \text{subject to} \quad & \mathcal{C}(y) \leq 0, \end{aligned} \tag{1}$$

where constraints are denoted by $\mathcal{C}(\cdot) \leq 0$, decision variables are denoted by $y \in \mathcal{R}^n$, and the vector containing the local weights of variables are denoted by $P$. In order to apply ILP to resolve conflicts from decisions of neural models, prior work (Rizzolo and Roth, 2016; Punyakanok et al., 2004; Ning et al., 2018; Guo et al., 2020) has defined $P$ to be the vector of raw probabilities of local decisions, $P = [p^1, ..., p^n]$, where $p^i$ corresponds to the probability generated from a certain model for the $i$th decision variable ($y_i$). The global inference is modeled such that the combination of probabilities subject to constraints is maximized.

Previous use of ILP has proven effective in ensuring decision consistency in certain cases (Faghihi et al., 2023b) but did not address model heterogeneity. This problem becomes more dominant in scenarios where output probabilities come from independent models, making them less directly comparable. To address this limitation, we extend the ILP formulation beyond just considering the raw model probabilities. Instead, we map these raw scores into globally comparable values, facilitating a more balanced global optimization. We achieve this by incorporating additional information, such as decision confidence, expected model accuracy, and estimated prior probabilities. While previous studies have explored the integration of uncertainty in modeling the training objective (Xiao and Wang, 2019; Gal and Ghahramani, 2016; Zhu and Laptev, 2017), our work represents a novel effort in systematically incorporating multiple factors of this nature into the inference process for interrelated decisions to leverage external knowledge effectively.

# 2 Method

Our objective is to devise an improved scoring system, generating new local variable weights (importance) $W$ in the ILP formulation. Thus, we modify the original objective function as follows:

$$\text{Maximize} \quad W^{\top} y, \tag{2}$$

where $W = [w^1, ..., w^n]$. To determine the new weights, we aim to find the scoring function $G$, which normalizes the local predictions of each model and maps them into globally comparable values. For each model $m$ with multi-class decisions, we denote the output probabilities after applying a SoftMax layer as $P_m \subset P$. The scoring function $G$ transforms these raw probabilities into new weights $W_m \subset W$ to indicate the importance of the variables within the ILP objective, i.e., $W_m = G(P_m, m)$. In this section, we explore different options for the function $G$ and provide an intuitive understanding of their rationale.

## 2.1 Prior Probability (Output Size)

To facilitate fair comparison among decisions with varying output sizes, we consider a normalization factor based on prior probabilities. For an $N$-class output, the prior probability for each label is $\frac{1}{N}$ (assuming uniform distribution). This implies an inherent disadvantage for decisions made in larger output spaces. Thus, we normalize the raw probabilities by dividing them by the inverse of their respective priors and define $G(P_m, m) = P_m \times N$.

## 2.2 Entropy and Confidence

The outputs generated from models often exhibit varying levels of confidence. While raw probabilities alone may adequately indicate the model's confidence in individual Boolean decisions, a more sophisticated approach is required for assessing the models' confidence in multi-classification. We propose incorporating the entropy of the label distribution as an additional factor to assess the model's decision-making confidence. As lower entropy corresponds to higher confidence, we use the reverse of the entropy, normalized by the output size $N$, as a factor in forming the decision weight function $G(P_m, m) = P_m * (\frac{N}{Entropy(P_m)})$.

## 2.3 Expected Models' Accuracy

Assigning higher weights to the probabilities generated by more accurate models aligns the optimal solution with the overall underlying models' performance. This approach mitigates the influence of poor-quality decisions, which can negatively impact others in the global setting. We define the decision weight function $G$ as $G(P_m, m) = P_m * Acc_m$, where $Acc_m$ represents the accuracy of the corresponding model, measured in isolation. To mimic the real-world settings where test labels are not available during inference, we utilize the models' accuracies on a probe/dev set.

# 3 Empirical Study

We assess the impact of integrating proposed factors into the ILP formulation on a series of structured prediction tasks. Our approach is particularly suited for hierarchical structures encompassing multiple classes at different granularity levels, such as classical hierarchical classification problems. Additionally, we are the first to investigate the influence of enforcing global consistency on the procedural reasoning task, a complex real-world problem. To implement our method, we rely on the DomiKnowS framework (Rajaby Faghihi et al., 2021), offering a versatile platform that enables implementing and evaluating techniques to leverage external logical knowledge with minimal effort on structured output prediction tasks.

## 3.1 Metrics and Evaluation

We compare our method against two inference-time approaches: sequential decoding and basic ILP (ILP without our refinement). In contrast to ILP, sequential decoding, which relies on expert-designed rules or programs to enforce consistency, is unique to each dataset. In addition to conventional metrics (e.g., accuracy/F1), we include measurements that evaluate changes applied by the inference techniques: (1) total changes (**C**), (2) the percentage of incorrect-to-correct changes (**+C**), (3) the percentage of correct-to-incorrect changes (**-C**). We further evaluate all the baselines and inference methods on (1) the percentage of decisions satisfying task constraints and (2) Set Correctness, the percentage of correct sets of interrelated decisions (i.e., predictions of all levels in the hierarchy must be correct for an image). More details are in Appendix B.

## 3.2 Tasks

### 3.2.1 Procedural Reasoning

**Task:** Procedural reasoning task entails the tracking of entities within a narrative. Following Faghihi and Kordjamshidi (2021), we formulate this task as Question-Answering (QA). Two key questions are addressed for each entity $e$ and step $i$: (1) *Where is $e$ located in step $i$?* and (2) *What action is performed on $e$ at step $i$?*. The decision output of this task exhibits heterogeneity, encompassing a diverse range of possible actions (limited multi-class) and varied locations derived from contextual information (spans). The task constraints establish relationships between action and location decisions as well as among action decisions at different steps. For instance, the sequence of 'Destroy, Move' represents an invalid assignment for action predictions at steps $i$ and $i+1$.

**Dataset:** We utilize the **Propara** dataset (Dalvi et al., 2018), a small dataset focusing on natural events. This dataset provides annotations for involved entities and their corresponding location changes. The label set is further expanded to include information on actions, which can be inferred from the sequence of locations.

**Baseline:** We employ a modified version of the MeeT architecture (Singh et al., 2023) as our baseline for this task. The MeeT model is designed to ask the two aforementioned questions at each step and employs a generative model (T5-large) to answer those questions. The **Sequential Decoding** baseline resolves action inconsistencies in a sequential stepwise manner (first to last), followed by the selection of locations accordingly. Additional information can be found in Appendix A

### 3.2.2 Hierarchical Classification

**Task:** This task involves classifying inputs into various categories at distinct levels of granularity, establishing parent-child relationships between the classes where those follow a hierarchical structure.

**Datasets:** We employ three different datasets. (1) A subset of the Flickr dataset (Young et al., 2014) with two hierarchical levels for the classification of images with types of *Animal, Flower, and Food*, (2) 20News dataset for text classification, where the label set is divided into two levels, and (3) The OK-VQA benchmark (Marino et al., 2019), a subset of the COCO dataset (Lin et al., 2014). In OK-VQA, the hierarchical relations between labels are established into four levels based on ConceptNet triplets and the dataset's knowledge base.

**Baselines:** ResNet (He et al., 2016) and BERT (Devlin et al., 2019) are used to obtain representations for the image and text modalities, respectively. Linear classification layers are applied to convert obtained representations into decisions. The **Sequential Decoding** is top-down, bottom-up, and a two-stage (1) top-down on 'None' values and (2) bottom-up on labels for Animal/Flower/Food, 20 News, and VQA tasks, respectively. More information is available in Appendix A.

## 3.3 Results

Tables 1, 2, and 3 display results for *Animal/Flower/Food*, *Ok-VQA*, and *Propara* datasets. Due to space constraints, results for the *20News* dataset are in Appendix A.2. For close results, we use multiple seeds to validate reliability. Across experiments, the basic ILP technique favors decisions in smaller output spaces due to higher probability magnitudes (e.g., more changes in Actions than Locations in Table 3). Our new proposed variations can effectively mitigate this problem and perform a more balanced optimization.

**Animal/Flower/Food:** The sequential decoding establishes that the enforcement of the decisions originating from a model with better accuracy and with a smaller output size (Level 1) on other decisions may even have a negative impact on them (Level 2). In such scenarios, the inclusion of *Expected Accuracy* favors dominant decisions and adversely affects performance. However, the inclusion of *Prior Probability* proves effective in achieving a balanced comparison among decisions. In this task, despite the basic ILP formulation being detrimental, some of the new variations can even surpass the original baseline performance.

**Ok-VQA:** The baseline exhibits lower accuracy in lower-level decisions with smaller output sizes.

| Model | Level 1 (3) | | | | Level 2 (15) | | | | Average |
|---|---|---|---|---|---|---|---|---|---|
| | Acc | C | + C | - C | Acc | C | + C | - C | Acc |
| Baseline | 86.12 | - | - | - | **54.85** | - | - | - | 70.48 |
| Sequential | 86.12 | - | - | - | 54.39 | 32 | **15.625** | 37.5 | 70.25 |
| ILP | 86.07 | 16 | 43.75 | 43.75 | 54.43 | 16 | 12.5 | 37.5 | 70.25 |
| + Acc | 86.14 | 3 | 33.33 | **33.33** | 54.41 | 29 | 13.79 | 37.93 | 70.27 |
| + Prior | 86.30 | 24 | 50 | 41.67 | 54.78 | 8 | 12.5 | **25** | 70.54 |
| + Ent + Acc | 86.09 | 12 | 33.33 | 50 | 54.41 | 20 | 10 | 40 | 70.25 |
| + Ent + Prior | **86.42** | 25 | **52** | 40 | **54.82** | 7 | 14.29 | 28.57 | **70.62** |
| + All | 86.17 | 16 | 43.75 | 43.75 | 54.50 | 16 | 12.5 | 37.5 | 70.33 |

Table 1: Results on *Animal/Flower/Food* dataset on four random seeds. Reported values are the average scores of runs with close variances for all techniques (Level1: $\pm 1.6$ and Level2: $\pm 0.5$). **C** values are derived from the best run. $n$ in **Level** ($n$) denotes the number of output space classes. **Prior:** Prior Probability, and **Ent:** Entropy.

| Model | Level 1 (274) | Level 2 (158) | Level 3 (63) | Level 4 (8) | Average |
|---|---|---|---|---|---|
| Baseline | 56.73 | 54.45 | 43.43 | 17.68 | **54.64** |
| Sequential | 55.81 | 53.17 | 43.44 | 24.18 | 53.72 |
| ILP | 52.38 | 46.33 | **49.66** | **28.43** | 50.17 |
| + Acc | 55.65 | **54.67** | 48.15 | 23.73 | 54.23 |
| + Prior | 56.35 | 53.36 | 48.11 | 23.86 | 54.54 |
| + Ent + Acc | 56.43 | 53.25 | 48.1 | 24.02 | 54.56 |
| + Ent + Prior | 56.79 | 52.93 | 47.53 | 23.75 | **54.61** |
| + All | **56.84** | 52.66 | 46.98 | 22.63 | 54.5 |

Table 2: The results on the Ok-VQA dataset. The values represent the F1 measure. Levels 2, 3, and 4 contain 'None' labels. The low F1 measure of lower levels is due to a huge number of False Positives.

When applying the basic ILP method under these circumstances, a significant decline in results is observed, even below that of sequential decoding. However, incorporating any of our proposed factors leads to substantial improvements compared to the basic ILP formulation (over $4\%$ improvement) and can surpass the performance of sequential decoding. Particularly, combining *Entropy* and *Prior Probability* achieves the best performance. Notably, although the baseline model has higher overall performance, its inconsistent outputs are unreliable for determining the object label (see Table 4).

**Propara:** This is an example of a real-world task that involves hundreds of constraints and thousands of variables when combining decisions across entities and steps. Once again, basic ILP and *Expected Accuracy* factor prioritize decisions from the smaller output size (Actions). However, the *Prior probability* factor enables a more comparable space for resolving inconsistencies. Notably, the higher baseline performance is attributed to inconsistencies and cannot be used when reasoning about the process (See Table 4).

| Model | Actions (6) | | | | Locations (*) | | | | Average |
|---|---|---|---|---|---|---|---|---|---|
| | Acc | C | + C | - C | Acc | C | + C | - C | Acc |
| Baseline | **73.05** | - | - | - | 68.21 | - | - | - | **70.47** |
| Sequential | 71.56 | 75 | 13.33 | 46.66 | 67.63 | 255 | 27.8 | 32.2 | 69.47 |
| ILP | **73** | 63 | **36.5** | 38.1 | 66.38 | 217 | 19.8 | 35.9 | 69.47 |
| + Acc | **73** | 63 | **36.5** | 38.1 | 66.43 | 217 | 19.8 | 35.9 | 69.50 |
| + Prior | 72.88 | 119 | 31.93 | **34.45** | 67.54 | 138 | 23.2 | 32.6 | **70.03** |
| + Ent + Acc | 72.93 | 63 | 34.92 | 38.1 | 66.38 | 219 | 19.6 | 35.6 | 69.44 |
| + Ent + Prior | 71.62 | 209 | 25.83 | 37.32 | 68.16 | 53 | 26.4 | 28.3 | 69.78 |
| + All | 71.74 | 198 | 25.75 | 36.86 | **68.27** | 72 | **29.2** | **27.8** | 69.89 |

Table 3: Results on Propara dataset. The dataset comprises 1910 location decisions and 1674 action decisions. *The output size of location decisions depends on the context of each procedure.

| Dataset | Model | Satisfaction | Set Correctness |
|---|---|---|---|
| Animal/Flower | Baseline | 96.4 | 53.40 |
| | Sequential | 100 | **54.50** |
| | Ent + Prior | 100 | **54.50** |
| VQA | Baseline | 38.99 | 54.43 |
| | Sequential | 100 | 57.11 |
| | Ent + Prior | 100 | **58.92** |
| Propara | Baseline | 45.12 | 23.30 |
| | Sequential | 100 | 28.81 |
| | Prior | 100 | **30.93** |

Table 4: Results of our proposed technique, baselines, and expert-written decoding strategies in terms of constraint satisfaction and set correctness. The **Set Correctness** metric reflects the practical usability of sets of dependent decisions in downstream applications.

**Constraints:** Table 4 presents the results of satisfaction and set correctness metrics across various datasets. It is evident that our newly proposed method significantly outperforms the baseline in both of these metrics. Notably, the degree of improvement in set correctness is more pronounced when the initial consistency of the baseline is lower. This observation underscores the substantial significance of our proposed technique in ensuring the practical utility of model decisions in downstream applications by substantially increasing the proportion of correct interrelated decision sets. Furthermore, in comparison to sequential decoding, our proposed solutions demonstrate even greater performance enhancements, particularly in scenarios where the task complexity is higher, and global inference can exert its maximum effectiveness.

## 4 Conclusion

This paper introduced an approach for taking into account the uncertainty and confidence measures, including the decisions' prior probability, entropy, and expected accuracy, alongside raw probabilities when making globally consistent decisions based on diverse models. Through experiments on four datasets, we demonstrated the effectiveness of incorporating our idea within the ILP formulation. This contribution represents a significant advancement in integrating large models in a unified decision-making framework for conducting complex tasks requiring interrelated decisions.

## Limitations

Our implementation of Integer Linear Programming (ILP) is based on the DomiKnowS framework, which relies on the Gurobi optimization engine (Gurobi Optimization, LLC, 2023). The availability of the Gurobi optimization engine in its free version is limited, which may pose constraints on the replication of our ILP-based approach for procedural reasoning experiments. However, the free academic license for Gurobi ensures the necessary access to execute all the tasks modeled in this paper. It is important to note that while our experiments and discussions demonstrate the effectiveness of our proposed approach in addressing challenges encountered with conventional ILP utilization, it is not guaranteed to consistently yield improved performance in scenarios where the decision space of variables is already comparable or consists solely of boolean decisions. These limitations highlight the need for careful consideration and evaluation of the specific problem domain and characteristics when applying our approach or considering alternative methodologies.

## References


Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *Proceedings of the 6th International Conference for Learning Representations (ICLR)*.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2012. Structured learning with constrained conditional models. *Machine learning*, 88(3):399–431.

Daniel Dahlmeier and Hwee Tou Ng. 2012. A beam-search decoder for grammatical error correction. In *EMNLP 2012*, pages 568–578.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. 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 1595–1604.

Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. In *(EMNLP-IJCNLP)*, pages 4496–4505.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hossein Rajaby Faghihi and Parisa Kordjamshidi. 2021. Time-stamped language model: Teaching language models to understand the flow of events. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4560–4570.

Hossein Rajaby Faghihi, Parisa Kordjamshidi, Choh Man Teng, and James Allen. 2023a. The role of semantic parsing in understanding procedural text. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1792–1804.

Hossein Rajaby Faghihi, Aliakbar Nafar, Chen Zheng, Roshanak Mirzaee, Yue Zhang, Andrzej Uszok, Alexander Wan, Tanawan Premsri, Dan Roth, and Parisa Kordjamshidi. 2023b. Gluecons: A generic benchmark for learning under constraints. *arXiv preprint arXiv:2302.10914*.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. *ACL 2017*, page 56.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Quan Guo, Hossein Rajaby Faghihi, Yue Zhang, Andrzej Uszok, and Parisa Kordjamshidi. 2021. Inference-masked loss for deep structured output learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2754–2761.

Quan Guo, Hossein Rajaby Faghihi, Yue Zhang, Andrzej Uszok, and Parisa Kordjamshidi. 2020. Inference-masked loss for deep structured output learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2754–2761. International Joint Conferences on Artificial Intelligence Organization. Main track.

Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on* *computer vision and pattern recognition*, pages 770–778.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *54th ACL*, pages 2410–2420.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2023. Transferring procedural knowledge across commonsense tasks. *arXiv preprint arXiv:2304.13867*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Tianyu Liu, Yuchen Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022. Autoregressive structured prediction with language models. *arXiv preprint arXiv:2210.14698*.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204.

Yatin Nandwani, Abhishek Pathak, Parag Singla, et al. 2019. A primal dual formulation for deep learning with constraints. In *Advances in Neural Information Processing Systems*, pages 12157–12168.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1346–1352.

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*, 21:1–67.

Hossein Rajaby Faghihi, Quan Guo, Andrzej Uszok, Aliakbar Nafar, and Parisa Kordjamshidi. 2021. DomiKnowS: A library for integration of symbolic domain knowledge in deep learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 231–241, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Nick Rizzolo and Dan Roth. 2016. Integer linear programming for coreference resolution. *Anaphora Resolution: Algorithms, Resources, and Applications*, pages 315–343.

Dan Roth and Wen-tau Yih. 2005. Integer linear programming inference for conditional random fields. In *Proceedings of the 22nd international conference on Machine learning*, pages 736–743.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. Picard: Parsing incrementally for constrained auto-regressive decoding from language models. In *EMNLP*, pages 9895–9901.

Janvijay Singh, Fan Bai, and Zhen Wang. 2023. Entity tracking via effective use of multi-task learning model and mention-guided decoding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1255–1263, Dubrovnik, Croatia. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. Deepstruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823.

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7322–7329.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. 2018. A semantic loss function for deep learning with symbolic knowledge. In *ICML*, pages 5502–5511. PMLR.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Lingxue Zhu and Nikolay Laptev. 2017. Deep and confident prediction for time series at uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 103–110. IEEE.

## A Datasets & Baselines

### A.1 Animal/Flower/Food

The dataset[1] employed in this study is sourced from the online platform 'Flickr' and encompasses a total of 5439 images classified into three primary categories, namely 'Flower,' 'Animal,' and 'Food.' In the absence of an officially designated test set, a random partitioning strategy is adopted to ensure comparability in the distribution of training and testing instances. Consequently, the resulting splits are utilized within the experimental framework. The training subset encompasses 4531 images, while the test set comprises 1088 images. The dataset further comprises various sub-categories, including 'cat,' 'dog,' 'monkey,' 'squirrel,' 'daisy,' 'dandelion,' 'rose,' 'sunflower,' 'tulip,' 'donuts,' 'lasagna,' 'pancakes,' 'pizza,' 'risotto,' and 'salad.' It should be noted that the data distribution across labels is not balanced, posing a more challenging classification task. This dataset is employed as a simplified scenario to illustrate the benefits of the proposed inference approach.

As the baseline for this task, we use ResNet-50 to represent the images and add a single layer MLP on top for each level. The model is further trained by Cross-Entropy objective and AdamW as optimizer.

The sequential decoding strategy for this dataset propagates labels in a top-down manner, where the highest probable children of the selected Level1 decisions is chosen as the prediction at Level2.

### A.2 20News

This dataset comprises a collection of diverse news articles classified into 23 distinct categories. In order to capture the hierarchical structure inherent in the dataset's labels, we partition these categories into two levels. It should be noted that certain higher-level concepts lack corresponding lower-level labels, necessitating the inclusion of a 'None' label at level 2. Furthermore, we perform a removal process on the initially annotated data containing the 'None' labels, as this subset primarily consists of noisy documents that do not align with any categories present within the dataset. It is crucial to differentiate this removal process from the intentional addition of the 'None' label at level 2, which we manually introduced.

[1] https://github.com/kaustubh77/Multi-Class-Classification

| Model | Level 1 (16) | | | | Level 2 (8) | | | Average |
|---|---|---|---|---|---|---|---|---|
| | F1 | C | + C | - C | F1 | C | + C | F1 |
| Baseline | 73.62 | - | - | - | 75.13 | - | - | 74.01 |
| Sequential | 72.99 | 330 | 20.6 | 46.36 | 75.13 | 0 | 0.00 | 73.55 |
| ILP | 73.53 | 225 | 25.78 | 39.55 | 75.46 | 68 | 63.24 | 74.03 |
| + Acc | 73.57 | 212 | **26.89** | 39.62 | 75.45 | 73 | 64.39 | 74.05 |
| + Prior | 73.35 | 161 | 25.46 | 39.13 | 75.35 | 94 | 65.96 | 74.01 |
| + Ent + Acc | 73.54 | 205 | 26.34 | 40 | 75.39 | 75 | **64** | 74.02 |
| + Ent + Prior | 73.63 | 125 | 26.4 | 36 | 75.49 | 112 | 68.75 | 74.12 |
| + All | **73.64** | 131 | 25.95 | **35.11** | **75.52** | 111 | 68.47 | **74.13** |

Table 5: Results on 20News dataset. Here, the *-C* of level 2 is 0 in all cases.

As the baseline for this task, we initially employed the Bert-Base encoder to generate representations for each news story. Due to the limited context size of Bert, which is constrained to a maximum of 512 tokens, we truncate the news articles accordingly and utilize the CLS token as the representative embedding for the entire article. For Level 1, a 2-layer Multilayer Perceptron (MLP) architecture is employed, with LeakyReLU serving as the chosen activation function. Additionally, Level 2 decisions are made using a single-layer MLP. During the training process, the model is optimized using the AdamW optimizer, with the Cross-Entropy loss function being employed.

The sequential decoding strategy is this dataset is a bottom-up strategy. Here, the model's decision from Level2 is propagated into Level1 without looking further into the initial probabilities generated by the model at that level.

#### A.2.1 Results

The baseline performance is similar across different decisions. Thus, considering either the *Expected Accuracy* or the *Prior Probability* in isolation does not have a substantial impact on the global optimization process. However, the inclusion of all proposed factors *(Entropy, Accuracy, and Prior Probability)* leads to a balanced and optimal solution. Although the overall task performance in this experiment does not show significant improvements, this is mainly because the initial decision inconsistencies are minimal. Nevertheless, evaluating the positive and negative changes provides valuable insights into the significance of incorporating the proposed factors.

### A.3 OK-VQA (COCO)

The OK-VQA dataset is primarily introduced as a means to propose an innovative task centered around question-answering utilizing external knowledge. To construct this dataset, a subset of the COCO dataset is employed, with augmented annotations obtained through crowdsourcing. While the main objective of the dataset revolves around question answering, it is important to note that it encompasses two levels of annotation. These annotations not only indicate the answer to the given question but also provide additional clarifications regarding the types of objects depicted in the corresponding images. In order to leverage knowledge pertaining to image type relationships, the label set is expanded to include supplementary high-level concepts. Additionally, a knowledge base is provided, delineating parent-child relationships between these labels. The dataset comprises a total of 500 object labels. To enhance the breadth of knowledge encompassed by the dataset, we incorporate additional information from ConceptNet to establish comprehensive relationships among the labels. Notably, both the new information and the original knowledge base may contain noisy information. This, in conjunction with the original knowledge base, forms a four-level hierarchical dependency among the initial 500 labels. Consequently, certain labels within each level may not possess corresponding children at lower levels, necessitating the introduction of 'None' labels at levels 2, 3, and 4.

In this study, we employ the Faster R-CNN framework (Ren et al., 2015) along with ResNet-110 as the chosen methodology to represent individual objects within images. Subsequently, a one-layer Multilayer Perceptron (MLP) architecture is utilized to classify the images at each level of the hierarchical structure. It should be noted that the number of positive examples (i.e., labels that are not denoted as 'None') decreases as we move toward lower levels of the hierarchy. To address this, we perform subsampling on the 'None' labels for the corresponding classifiers at those levels. The models are trained with the Cross-Entropy loss function and the AdamW optimizer.

The sequential decoding strategy for this dataset is a two-stage top-down and then bottom-up process. Here, 'None' labels are first propagated from Level 1 to Level 4, and then the selected label (if not None) from Level 4 is propagated bottom-up to Level 1. Since each label at level$n$ only has one parent in Level$n-1$, this process does not need to look into the original model probabilities for propagation.

## A.4 Propara

The Propara dataset serves as a procedural reasoning benchmark, primarily devised to assess the ability of models to effectively track significant entities across a series of events. The stories within this dataset revolve around natural phenomena, such as photosynthesis. The annotation process involves capturing crucial entities and their corresponding locations at each step of the process, which are obtained through crowd-sourcing efforts. An illustrative example of this dataset is depicted in Figure 1.

The sequence of locations pertaining to each entity can be further extended to infer the actions or status of the entity at each step. Previous studies (Dalvi et al., 2019) have proposed six possible actions for each entity at each step, namely 'Create,' 'Move,' 'Exist,' 'Destroy,' 'Prior,' and 'Post.' In this context, 'Prior' signifies an entity that has not yet been created, while 'Post' denotes an entity that has already been destroyed.

| Process | Participants | | | |
|---|---|---|---|---|
| Sentences | plant | animal | bone | oil |
| Before the process begins | ? | ? | - | - |
| 1. Plants and animals die in a watery environment | watery environment | watery environment | - | - |
| 2. Over time, sediments build over | sediment | sediment | - | - |
| 3. The body decomposes | sediment | - | sediment | - |
| 4. Gradually buried material becomes oil | - | - | - | sediment |

Figure 1: An example from the Propara dataset taken from (Faghihi et al., 2023a). '-' refers to the entity not existing; '?' refers to the entity whose location is unclear.

As for the baseline, we employ a modified version of the MeeT (Singh et al., 2023) architecture. The architecture utilizes T5-Large (Raffel et al., 2020) as the backbone and employs a Question-Answering framework to extract the location and action of each entity at each step. The format of the input to the model is as follows for entity $e$ and step $i$: "Where is $e$ located in sent $i$? Sent 1: ..., Sent 2: ..., ...". For extracting the action, the set of options is also passed as input, resulting in the modification of the question to "What is the status of entity $e$ in sent $i$? (a) Create (b) Move (c) Destroy (d) Exist (e) Prior (f) Post".

Although the original model of MeeT incorporates a Conditional Random Field (CRF) (Lafferty et al., 2001) layer during inference to ensure consistency among action decisions, we exclude this layer from our baseline. This decision is motivated by two reasons. Firstly, the use of CRF in this context is not generalizable as it relies on training data statistics for defining transitional scores. Secondly, we intend to impose consistency using various inference mechanisms on our end and consider a joint framework to ensure both locations and actions exhibit consistency. Additionally, while the MeeT baseline employs two independent T5-Large models for each question type (location and action), our baseline utilizes the same model for both question types. For the sequential decoding technique to enforce sequential consistency among the series of interrelated action and location decisions, we utilize the post-processing code presented in Faghihi et al. (2023a).

# B Metrics

Here, we briefly describe the metrics used in this paper to evaluate the methods.

## B.1 Number of Changes

This metric quantifies the post-inference changes in decisions, specifically assessing the extent to which original decisions are altered due to inference constraints. It serves as a crucial indicator of whether the optimization method treats all decisions equally or exhibits a preference for certain decisions over others. A genuinely global optimization method will result in multiple decision changes, promoting a more balanced distribution of alterations across all decisions. In contrast, expert-written strategies tend to favor specific decisions. This metric is straightforward to calculate by comparing the differences between decisions before and after applying the inference mechanism.

## B.2 Ratio of In-Correct to Correct Changes (+C)

This metric reveals the proportion of post-inference changes that are deemed favorable. While this metric may not carry substantial standalone significance, it serves as a valuable means to compare different inference techniques. A higher ratio signifies that the inference method has been more successful in deducing accurate labels based on the imposed constraints.

## B.3 Ratio of Correct to In-Correct Changes (-C)

This number shows the extent of undesirable changes made after inference. A lower ratio means the inference method has done a better job of preventing errors while ensuring the output adheres to the constraints.

## B.4 Satisfaction Rate

This metric shows how well predictions align with constraints. We calculate it by generating constraint instances from related decisions and counting the satisfying cases against all possible instances. Inference techniques guarantee that modified decisions always adhere to the constraints, resulting in a satisfaction rate of 100%.

## B.5 Correctly Predicated Sets of Interrelated Decisions

This metric is crucial for assessing the practical usefulness of the output from inference techniques or the original network decisions in downstream applications. The primary objective of inference mechanisms is to boost the percentage of these fully satisfying cases compared to the model's original performance, all while ensuring that the decisions align with the task's constraints. For instance, in a hierarchical classification task, we consider one instance to be correct only when the decisions at all levels are simultaneously accurate.

# C Discussion

Here, we address some of the key questions about this work.

**Q1: Which metric is most important among the ones evaluated in this paper?**

All the metrics assessed in this paper provide insights into the model's performance. Among these, the **Set Correctness** score offers a comprehensive evaluation that combines constraint satisfaction and correctness, indicating the proportion of output decisions suitable for safe use in downstream tasks.

When comparing different ILP variations, the primary focus should be on the original task performance since they all share the same high satisfaction score of 100%. Additionally, the **Change** metric helps reveal whether an ILP variation conducts truly global optimization or exhibits a bias towards specific prediction classes.

In the context of comparing the baseline method with inference techniques, it is essential to consider both the **satisfaction** and **set correctness** scores. This is because the raw model predictions, as initially generated, may not be directly acceptable. For instance, if a model predicts a "Move" action for entity A at step 4, but the location prediction does not indicate a change in location, it becomes unclear whether entity A indeed changed locations or not.

**Why utilize the model's overall accuracy in the score function instead of its accuracy for a specific decision variable?**

In our context, we assume that each decision type corresponds to a specific model. Therefore, assessing the model's accuracy is the same as evaluating the accuracy of a particular decision type. If a single model supplies multiple decision types, we can easily expand this concept to evaluate the accuracy of each decision type individually within the same framework.

**What is the main difference between the sequential decoding strategy and the ILP formulation?** The sequential decoding strategy is a domain-specific, expert-crafted technique employed for addressing decision inconsistencies in accordance with task constraints. In contrast, the ILP (Integer Linear Programming) formulation offers a more general, non-customized approach that isn't tailored to individual tasks.

Sequential decoding strategies typically involve rules or programs that often exhibit a preference for a specific decision while adjusting other decisions to align with it. This approach tends to prioritize decision alignment over considering the probabilities associated with these decisions. On the other hand, the ILP optimization process seeks the most optimized solution by taking into account the raw probabilities from the models and the imposed constraints.