Neuro-Symbolic Action Anticipation with Learned Constraints

Muyang Yan[1], Maxwell J. Jacobson[1], Simon Stepputtis[2], Katia Sycara[2], and Yexiang Xue[1]

[1]Department of Computer Science, Purdue University {yan427, jacobs57, yexiang}@purdue.edu
[2]The Robotics Institute, Carnegie Mellon University {sstepput, sycara}@andrew.cmu.edu

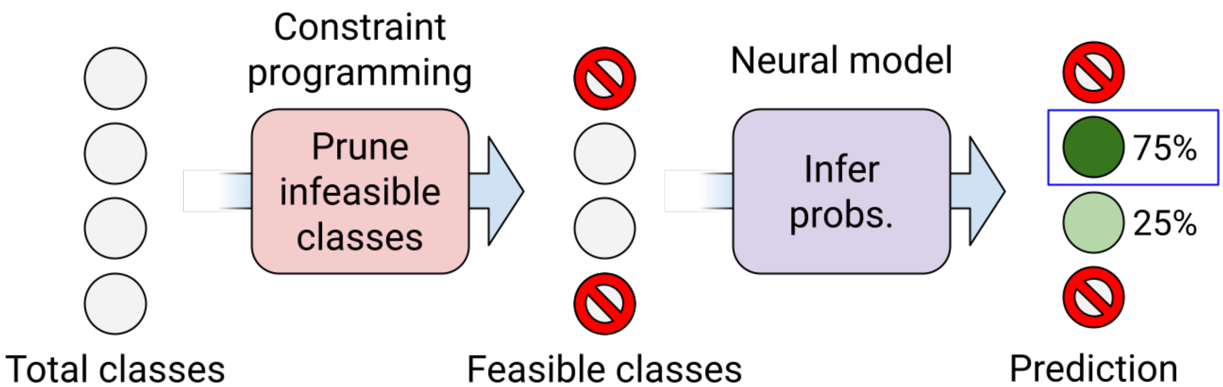

Figure 1: Our two-stage approach to integrating neural methods and and symbolic constraints for prediction tasks.

Machine Learning (ML) and Constraint Acquisition (CA)[1] are two leading paradigms for modeling data. ML learns an implicit model by searching a neural parameter space, while CA discovers explicit constraints consistent with the data by searching a constraint space. These approaches complement each other: ML is probabilistic and flexible but opaque, while CA is deterministic, rigid, and interpretable. In many applications, data arises through both deterministic and probabilistic processes. In these cases, combining ML and CA can produce a more suitable model than either paradigm on its own, while reaping the benefits of both.[4]

We demonstrate this combined approach in Action Anticipation, a vital task in areas like human-robot interaction, autonomous driving, and multi-agent systems. The goal is to predict an agent's future actions based on current and past observations of that agent. Past ML approaches use deep neural networks[3][5][7], thereby lacking guarantees on the feasibility of predicted actions. However, while the agents' actions are highly probabilistic, the actions that are feasible for an agent to perform in a given state are deterministic. For example, a chef can only cut a tomato if they are holding a knife.

Our method divides action anticipation into two stages: action pruning and action selection. We first apply explicit constraints learned through CA to prune infeasible actions based on the current state, then employ a neural network to select a final action from the feasible subset. Through this neuro-symbolic approach, we ensure predicted actions will be feasible. Our experiments show that using learned constraints outperforms neural methods in action pruning and improves overall action anticipation accuracy.

Experiments

We conducted experiments in a simulated environment. The environment is a grid-world consisting of a player, food items, and kitchen tools. Each grid tile is either walkable area or counter-top. The player can move, interact with objects, and use tools it picks up.

We represent the environment states as scene graphs, which have the benefit of being interpretable and thus easy to parse into logic. We use a Relational Graph Convolution Network[8] for neural prediction throughout our experiments and we do Constraint Acquisition using Popper[2], an Inductive Logic Programming[6] system.

Results

From data, we acquire the following constraints as requirements for each action verb to be feasible:

- Move: Agent near empty square
- Grab: Agent near counter, agent not holding anything, something grab-able on the counter.
- Put down: Agent holding something put-down-able.
- Cut: Agent holding knife, agent near counter, something uncut on top of counter

These rules accurately and concisely capture the conditions for feasibility, and are easily understood by humans.

Using our learned constraints, our methods outperform purely neural approaches in both action pruning and action anticipation.

Task	Method	EMR(%)	MAP(%)	MAR(%)
Action pruning	GNN	36	78.2	81.7
	Learned constraints (ours)	66	85.7	91.1
Action anticipation	No pruning, direct prediction w/ GNN	58	51.5	70.4
	Prune w/ GNN	61	48.9	52.5
	Prune w/ learned constraints (ours)	68	58.6	59.9

Table 1: Comparison of methods of action pruning and action anticipation. EMR, MAP and MAR stand for exact match ratio, mean average precision and mean average recall, respectively. We compare 2 methods of action pruning: multi-label prediction with a GNN, and applying our learned constraints. We compare 3 methods of action anticpation: directly predicting the actions with a GNN in one step, doing both pruning and selection with GNN's, and our method of pruning with our learned constraints and then selecting with a GNN.

Submission Details

Submitted to: Application Track.
Muyang, Max, and Yexiang will be attending. Muyang is an undergraduate student.

References

[1] Christian Bessiere, Frédéric Koriche, Nadjib Lazaar, and Barry O'Sullivan. Constraint acquisition. *Artificial Intelligence*, 244:315–342, March 2017.

[2] Andrew Cropper and Rolf Morel. Learning programs by learning from failures. *Machine Learning*, 110(4):801–856, April 2021.

[3] Dayoung Gong, Joonseok Lee, Manjin Kim, Seong Jong Ha, and Minsu Cho. Future transformer for long-term action anticipation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3052–3061, June 2022.

[4] Maxwell J. Jacobson and Yexiang Xue. Integrating symbolic reasoning into neural generative models for design generation. *Artificial Intelligence (AIJ)*, 2025. To appear.

[5] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10139–10149, October 2023.

[6] Stephen Muggleton and Luc de Raedt. Inductive Logic Programming: Theory and methods. *The Journal of Logic Programming*, 19-20:629–679, May 1994.

[7] Yan Bin Ng and Basura Fernando. Action Forecasting with Feature-wise Self-Attention, July 2021.

[8] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling Relational Data with Graph Convolutional Networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, pages 593–607, Cham, 2018. Springer International Publishing.