Like Father Like Son

By Raffaele Leo

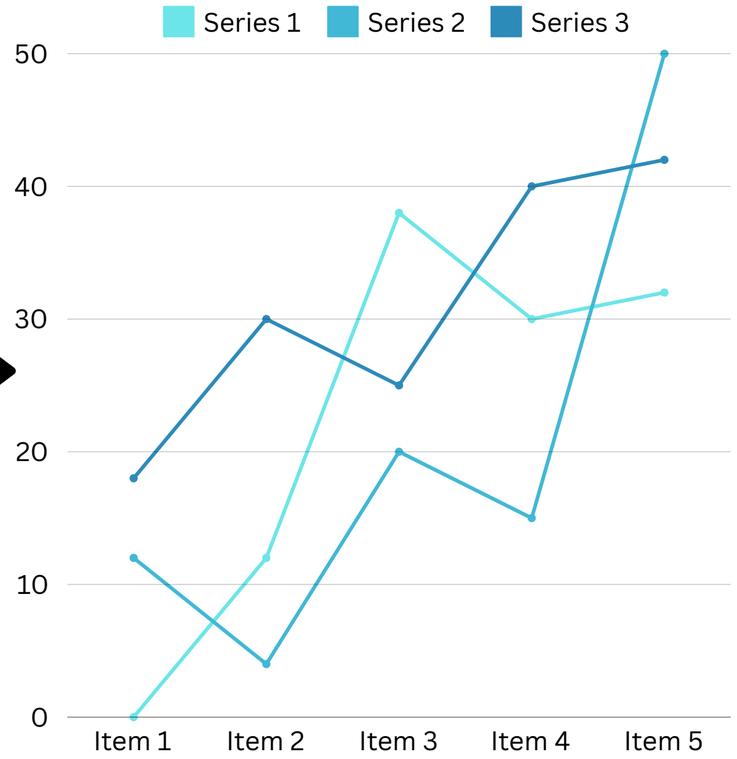
What is a Visualization Reccomendation System?

Query:

I wonder what the rate of adverbs vs verbs the author uses throughout each chapter

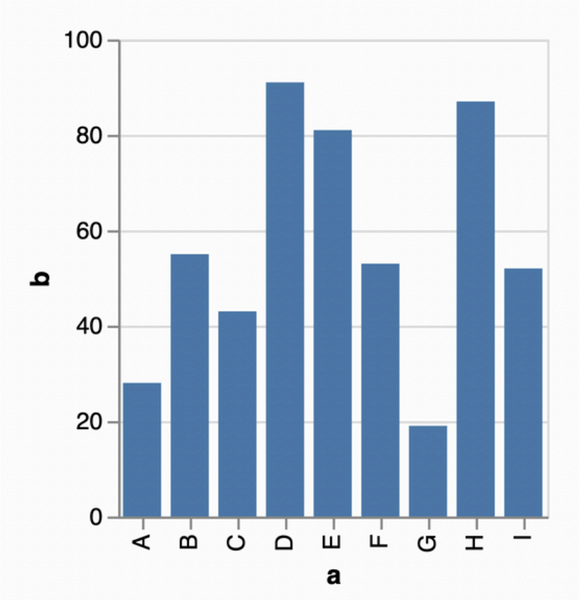
VRS:

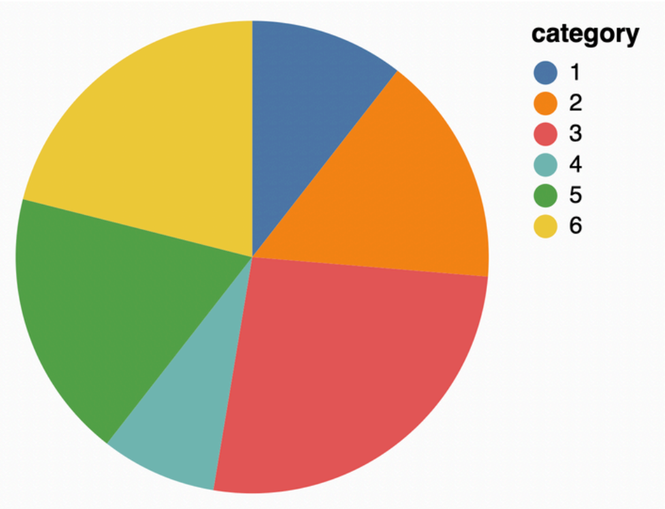
Output:

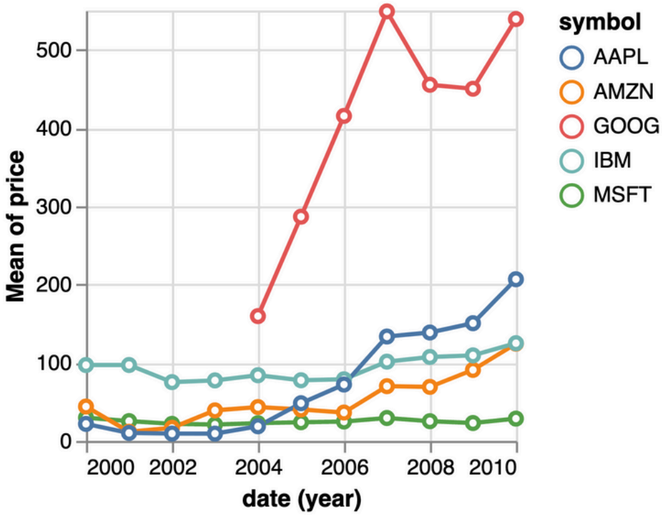


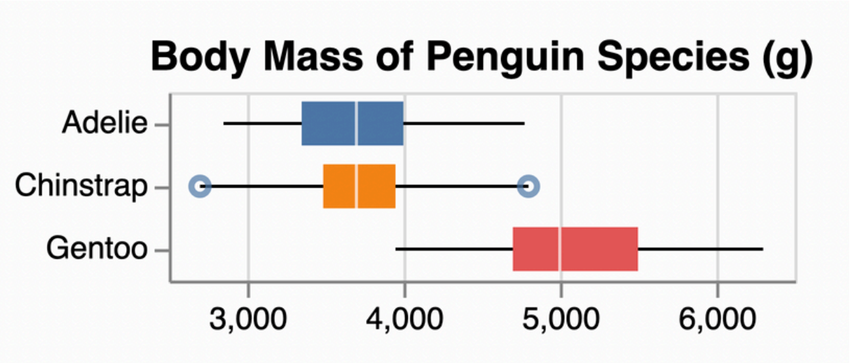
NVBench: Opening The Door for LLMs Within VRS

- 105 domains
- 7 types of visualizations
- 25,750 (NL, VIS) pairs



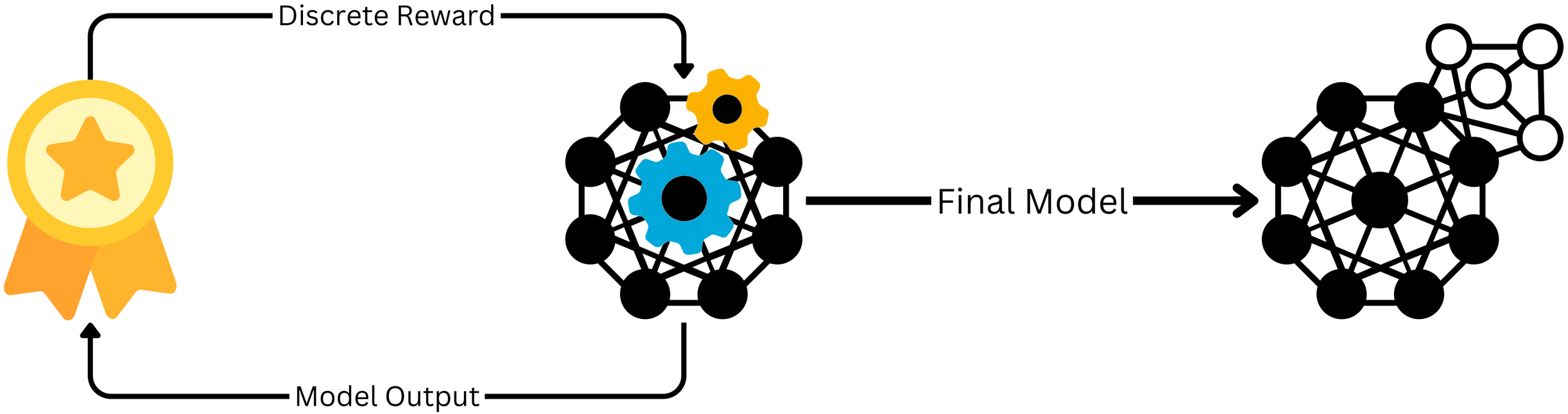






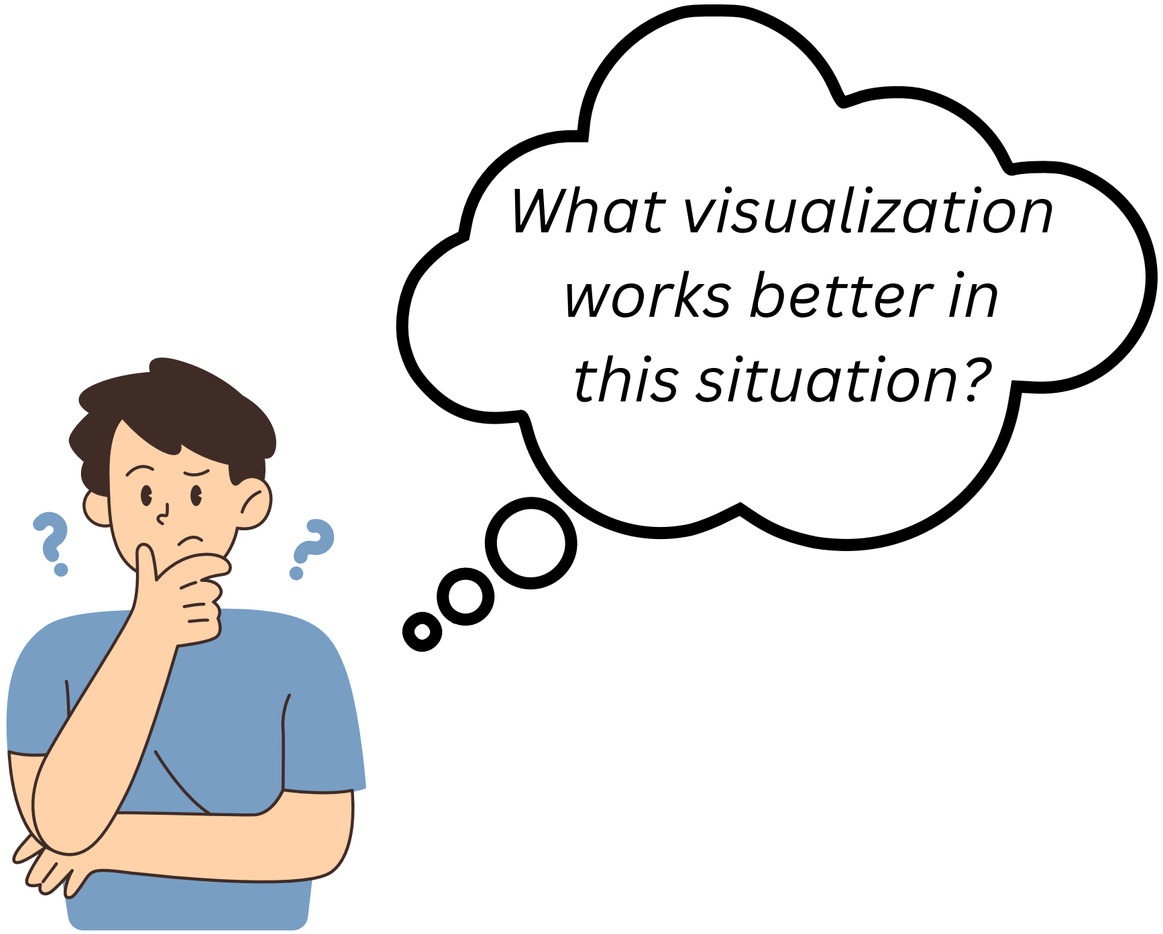
Reinforcement Learning Using Proximal Policy Optimization

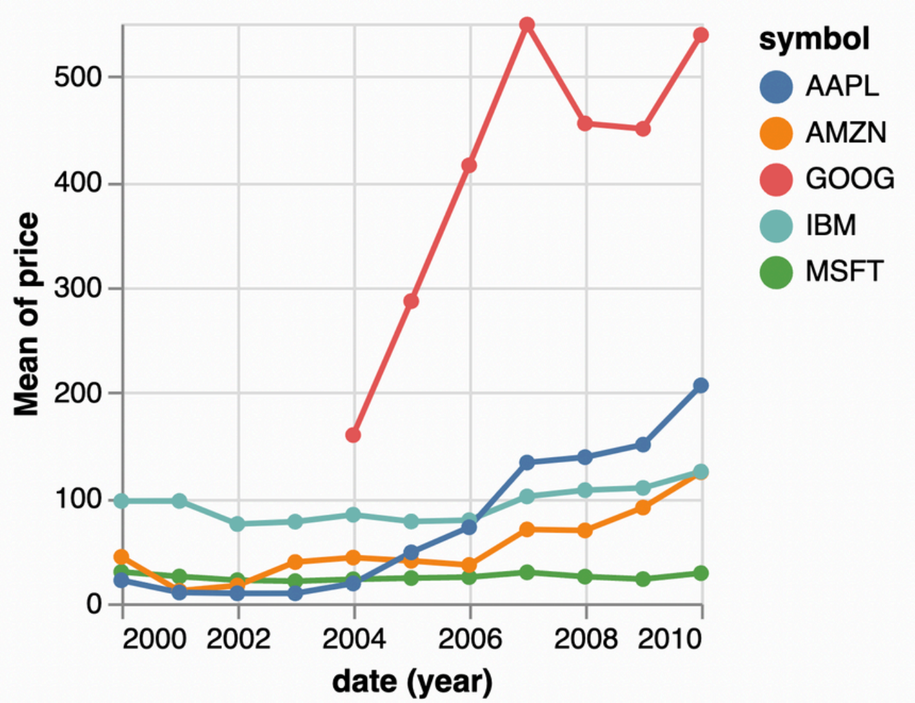
A more stable reinforcement learning method

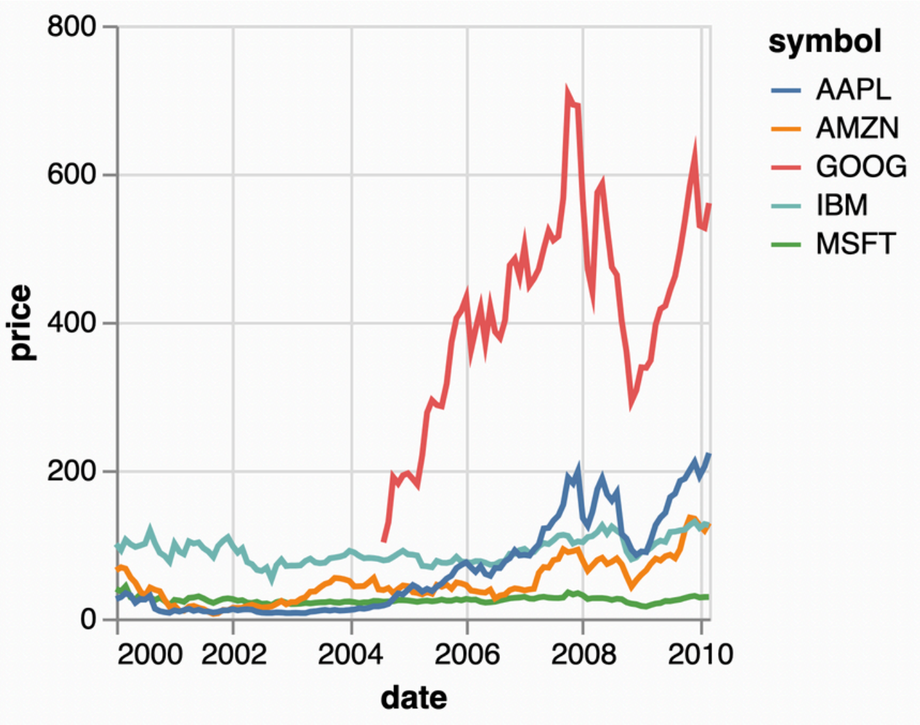


Problem Formulation

Using Reinforcement Learning to fine-tune visualization models is notoriously hard

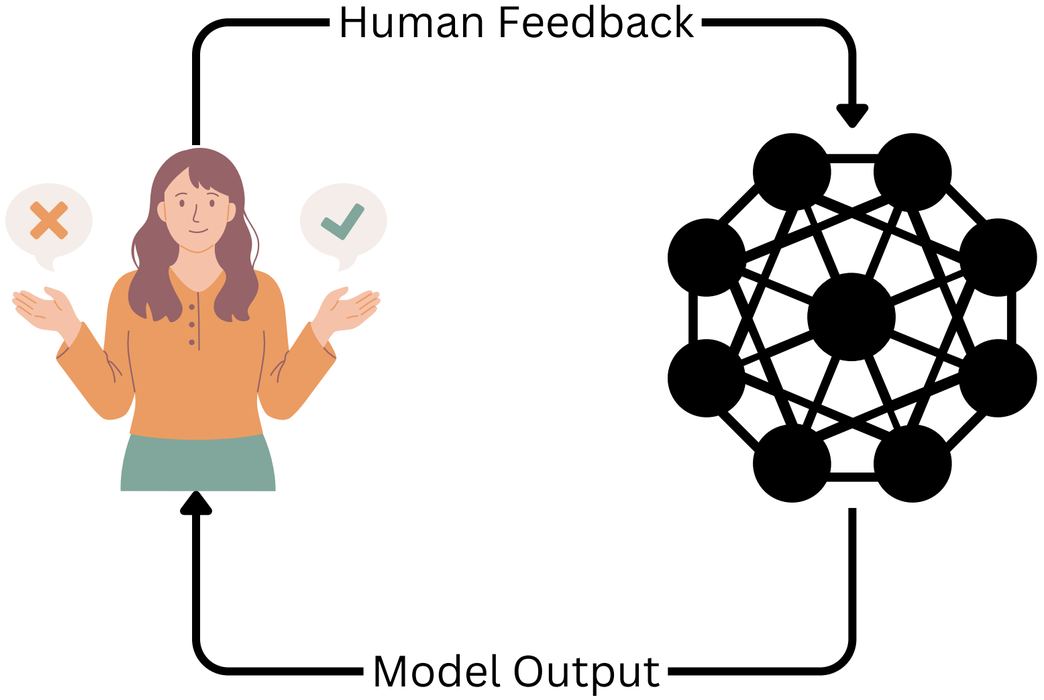






Current Solutions

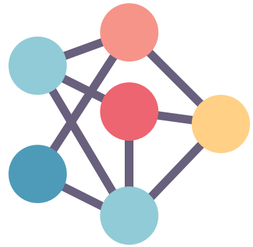
Reinforcement Learning With Human Feedback



Automated Reward Systems

Comparison Dataset

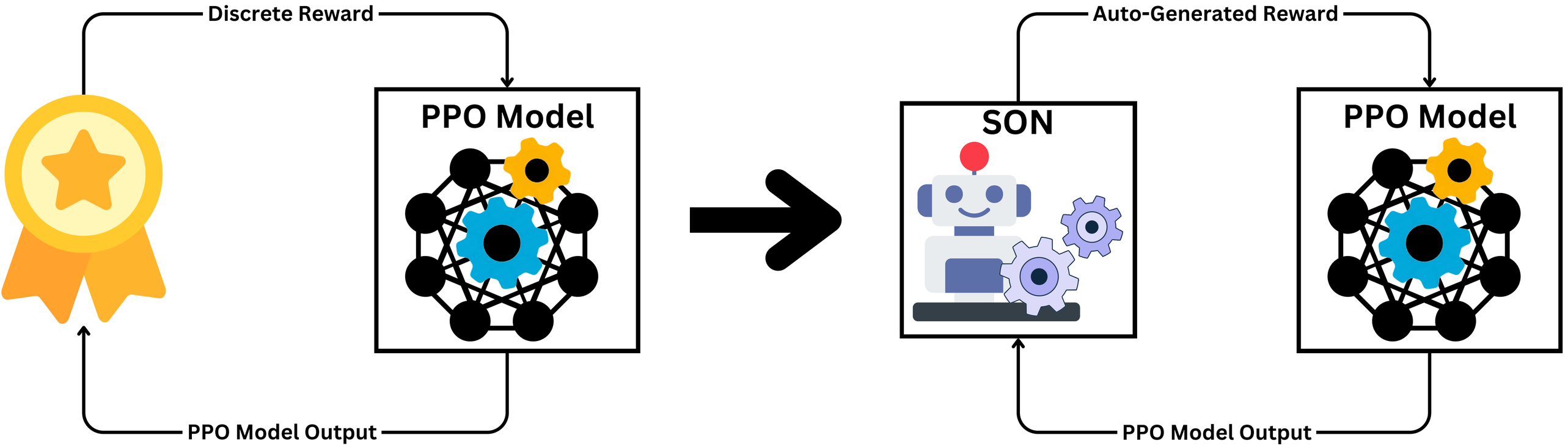
Reward Model



$$Loss = -log\left(\sigma\left(R_{Chosen} - R_{Rejected}\right)\right)$$

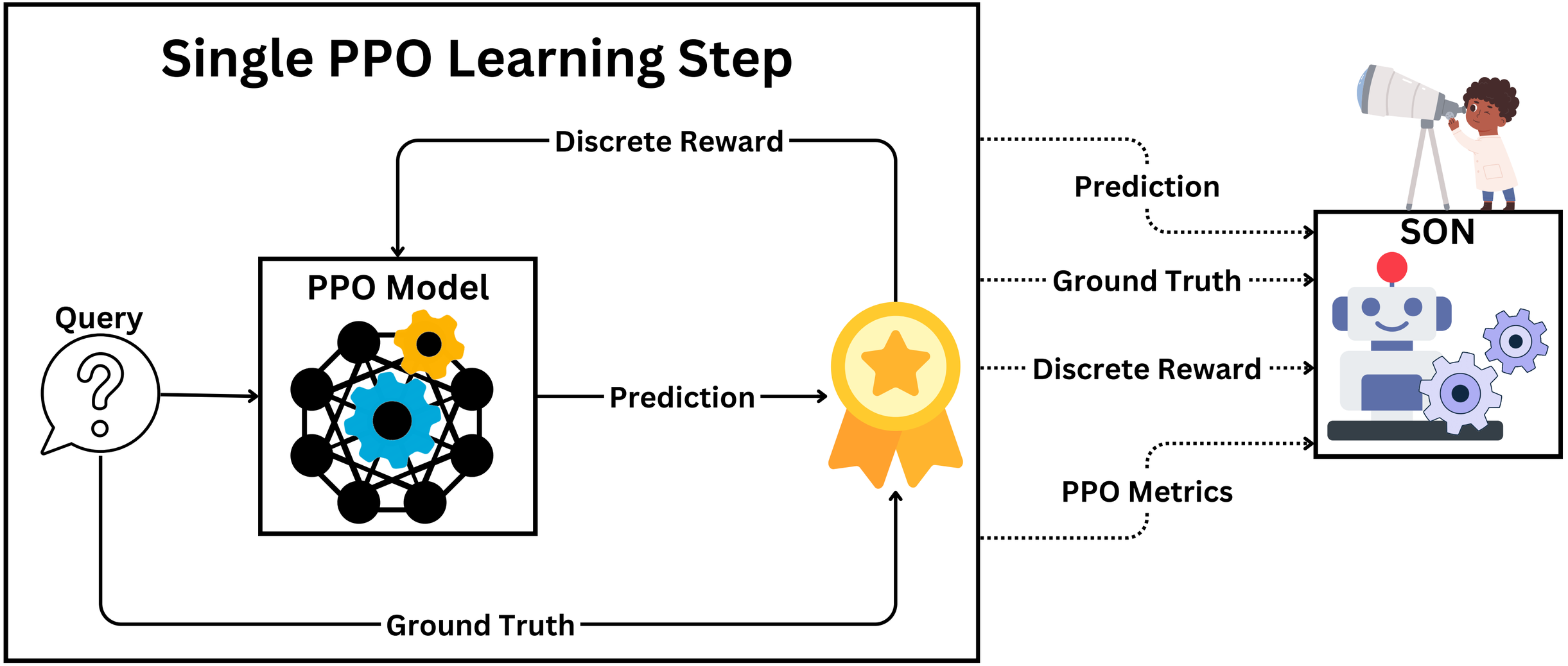
Stable Observer Network (SON)

Our solution to automatically generate rewards without needing a time consuming setup or an abundance of resources



Observation Phase

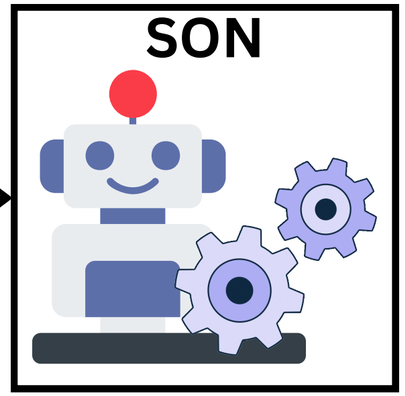
Our Stable Observer Network must first observe how the PPO model is responding to the discrete rewards using the PPO metrics.



Using PPO Feedback In the Loss Function

$$L = \alpha(R' - R) + \beta \frac{w_1 \Delta PL + w_2 \Delta VL}{(R' - R) + \epsilon}$$

R' — SON Predicted Reward →



PL (Policy Loss)

Measures how well the PPO Model is able to select rewards that maximize the overall expected value

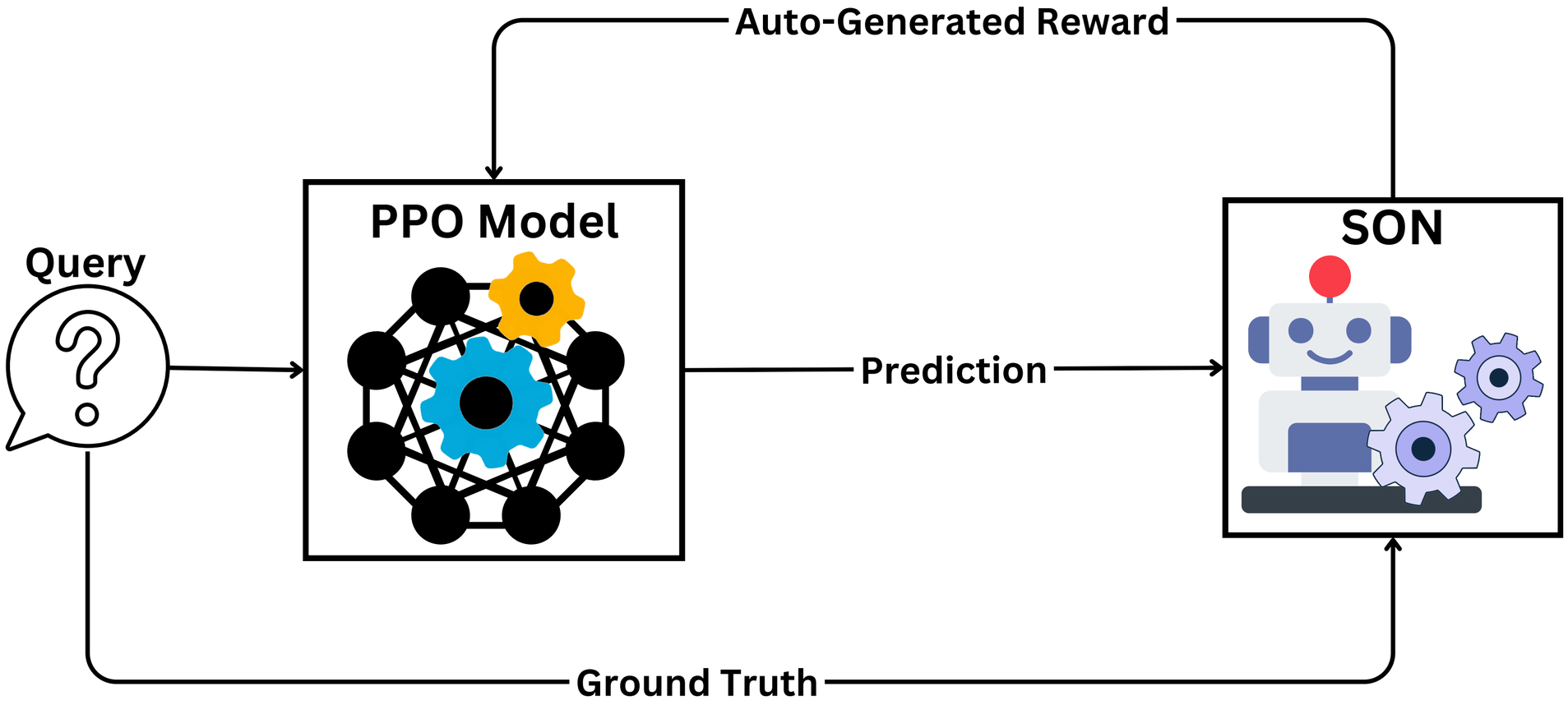
R —— Discrete Reward →

VL (Value Loss)

Measures how accurately the PPO Model is predicting the expected returns from a given state

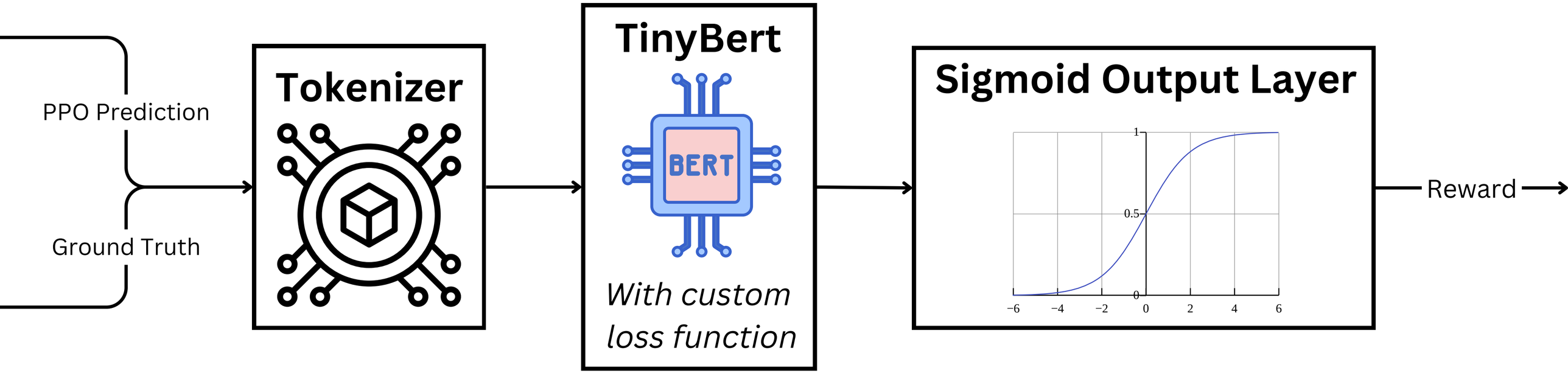
Inference Phase

After 200 iterations our SON has enough information about how the PPO model is learning and takes over as the sole reward model



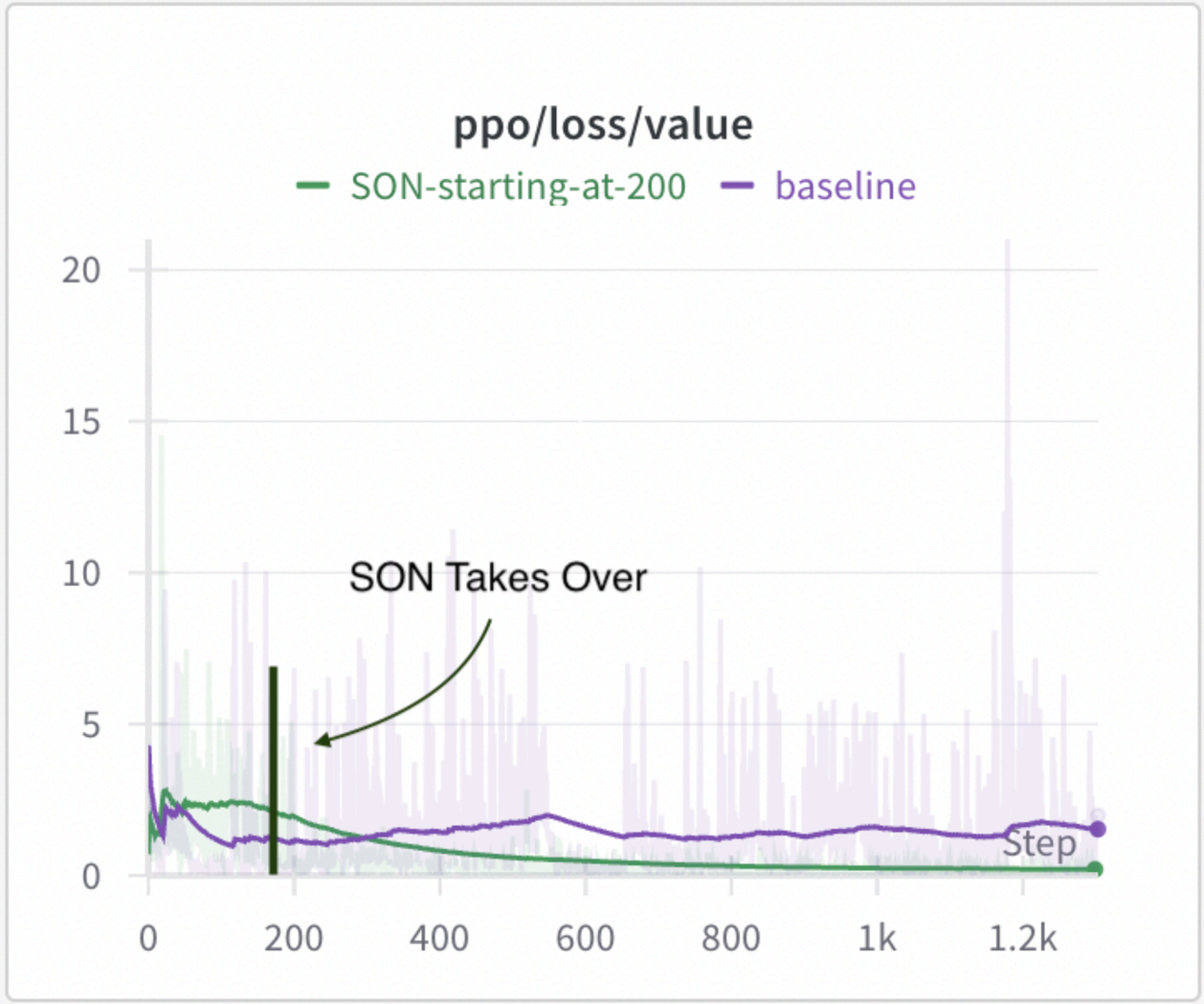
Structure of the SON

We customized the pre-trained TinyBert model, adding our custom loss along with a sigmoid function as the output layer



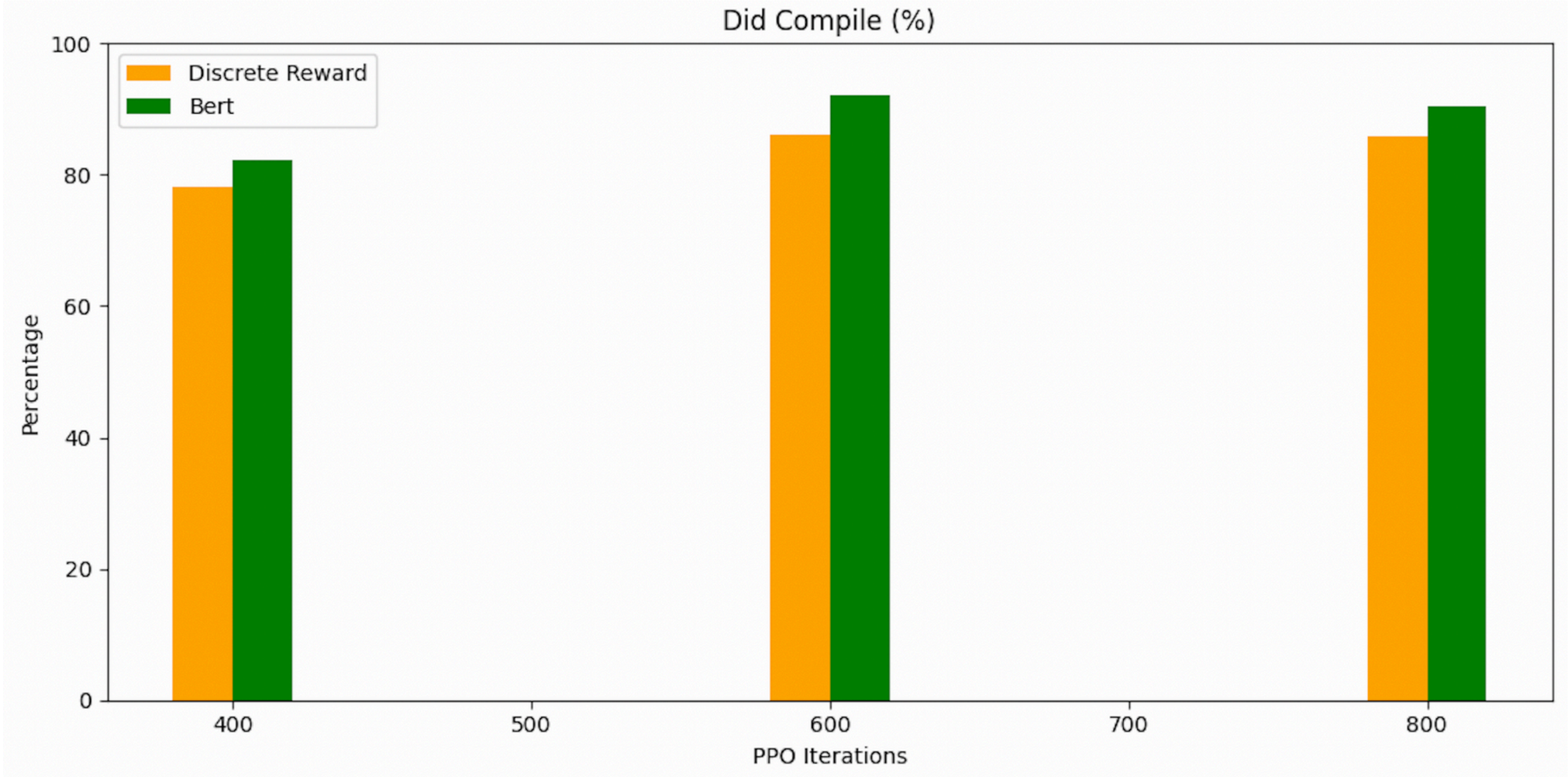
Proving The Hypothesis

When our SON takes over PPO learning becomes more stable



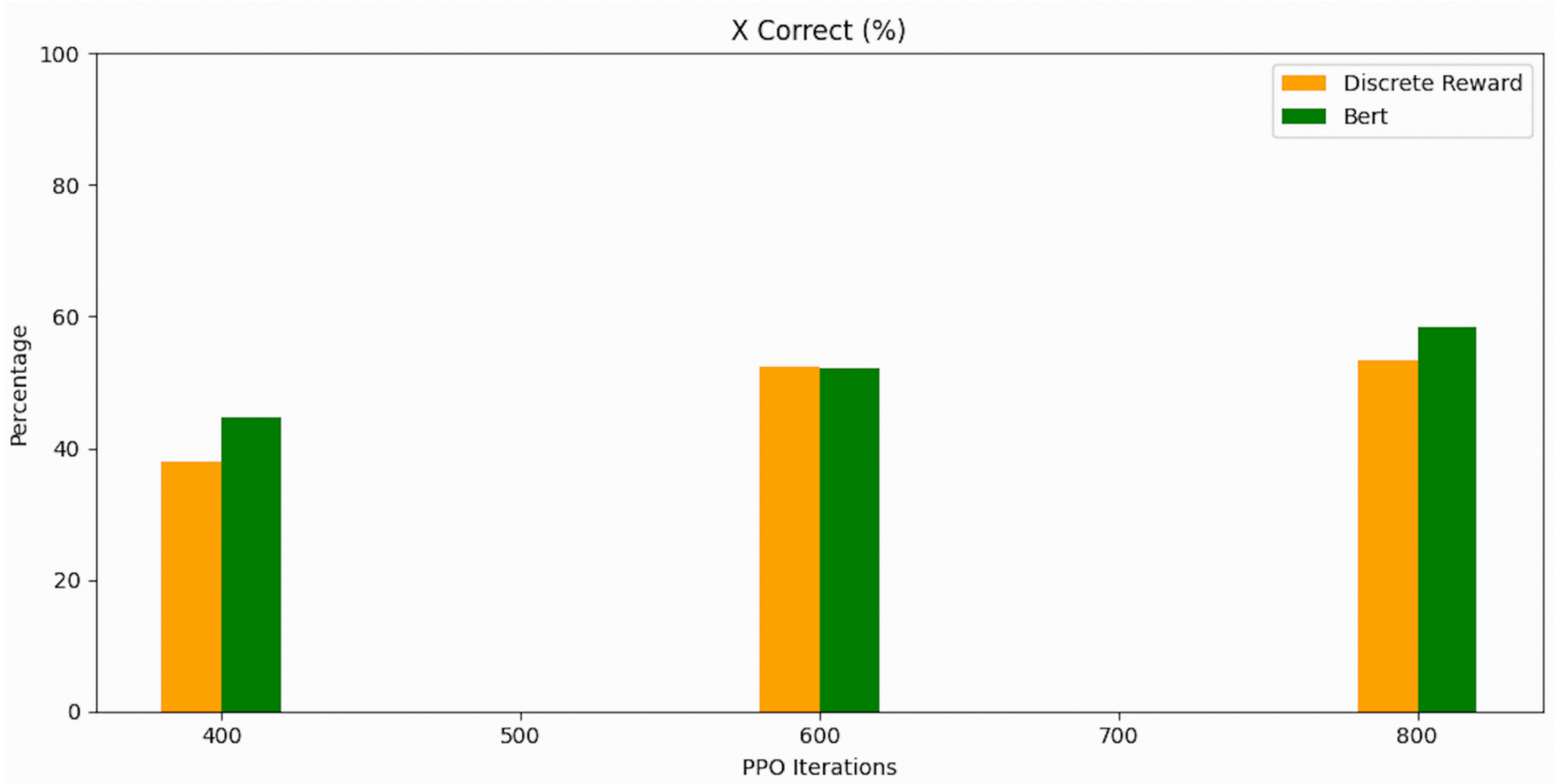
Results

This graph shows the number of times the PPO model's prediction compiled on our validation set, after being fine-tuned by a discrete reward (orange) and our SON (green)



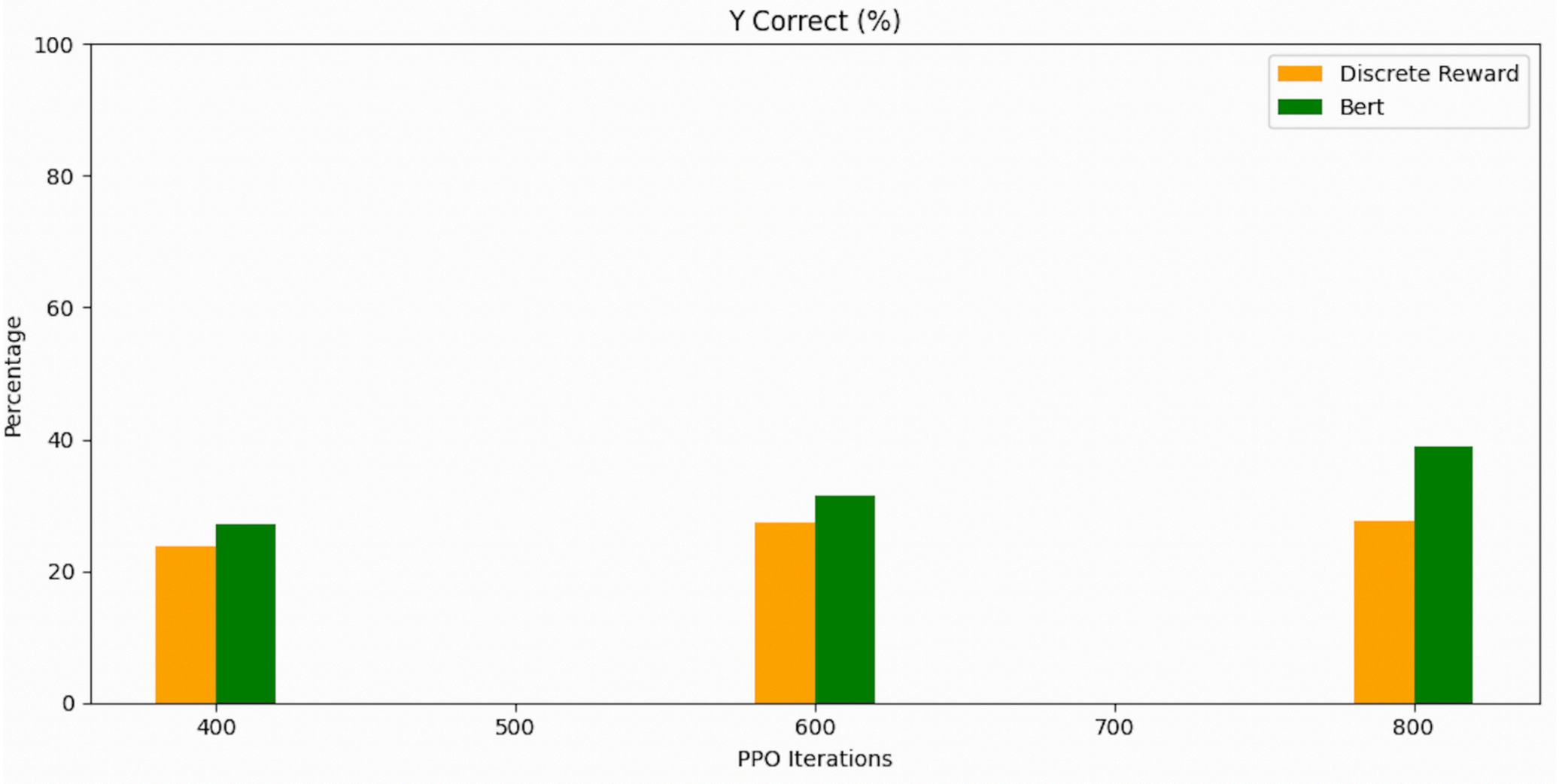
Results

This graph shows the number of times the PPO model's prediction got the x-axis correct on our validation set, after being fine-tuned by a discrete reward (orange) and our SON (green)



Results

This graph shows the number of times the PPO model's prediction got the y-axis correct on our validation set, after being fine-tuned by a discrete reward (orange) and our SON (green)

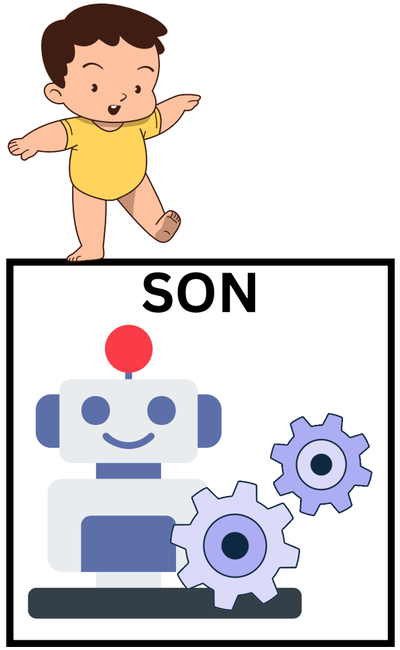


Sometimes We Just Need a Helping Hand

Using a Stable Observer Network we were able to:

- Auto-Generate rewards with little overhead
- Help the VRS avoid pitfalls while being fine-tuned
- Obtain faster and more stable convergence

Giving the PPO model a smaller model to ensure it stays on the right track seems to help it in the same way a fresh pair of eyes can help us solve a problem, or how a child can keep a parent focused on their goals.



Grazie Per La Vostra Attenzione