Combining LLM and RL and Logic Transformer

King-Yin Yan[0009−0007−8238−2442]

general.intelligence@gmail.com

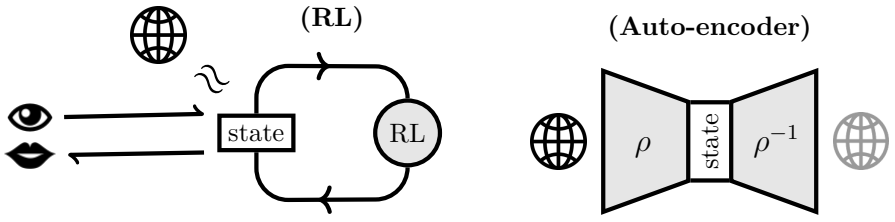
Abstract. blah

Keywords: AGI · large language models · reinforcement learning · neural-symbolic integration

1 LLM + RL architectures

For "string diagrams" there are usually two conventions: 1) data are nodes, functions are edges: $\textcircled{x} \xrightarrow{f} \textcircled{y}$ or alternatively 2) functions are nodes, data are edges: $\xrightarrow{x} \textcircled{f} \xrightarrow{y}$. In the following, I make explicit nodes for both functions (grey) and data (white), whereas edges merely represent linkages.

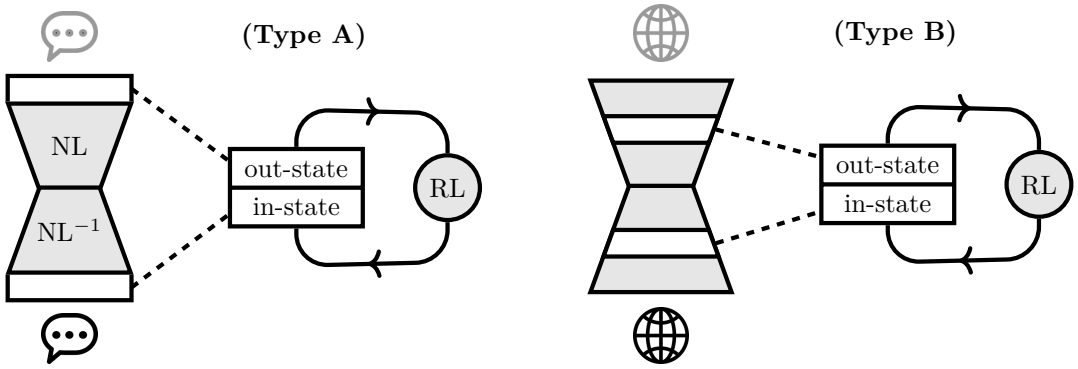
First we look at the **fundamental forms** of RL (left) and auto-encoders (right). The eye represents observations and the mouth (speech) actions. Because RL has to maximize rewards, its internal representation (the state) must eventually approach a good approximation of the world. The auto-encoder, of which LLMs are a special case, works by compressing world-data (via ρ) and de-compressing (via ρ^{-1}) to re-construct the data (grey world).



(1)

Next we look at two types of architectures for combining RL and LLM (cite). Type A is the "mainstream" approach, of which RLHF seems to be one instance.

Here, ... *The "format" of human internal thought is natural language.*



(2)