Latent Action Space for Efficient Planning in Theorem Proving

Minchao Wu*, Yuhuai Wu*

Australian National University / Data61 & Google / Stanford University

Motivation

Motivation

Action space of interactive theorem proving:

Motivation

Action space of interactive theorem proving:

- Unbounded action space

Motivation

Action space of interactive theorem proving:

- Unbounded action space
  - Commands of the prover (e.g., tactics with arguments)

Motivation

Action space of interactive theorem proving:

- Unbounded action space
  - Commands of the prover (e.g., tactics with arguments)
    - Theorem names
    - Arbitrary terms of the background logic of the proof system
    - Locations of application or substitution

Motivation

Action space of interactive theorem proving:

- Unbounded action space
    - Commands of the prover (e.g., tactics with arguments)
        - Theorem names
        - Arbitrary terms of the background logic of the proof system
        - Locations of application or substitution
- Computationally inefficient when planning

Motivation

Action space of interactive theorem proving:

- Unbounded action space
  - Commands of the prover (e.g., tactics with arguments)
    - Theorem names
    - Arbitrary terms of the background logic of the proof system
    - Locations of application or substitution
- Computationally inefficient when planning
  - Autoregressive generation

Contribution

In this work we

- learn a high-level representation of actions by embedding the raw action space into a latent action space, and

Contribution

In this work we

- learn a high-level representation of actions by embedding the raw action space into a latent action space, and
- learn a world model in the latent space for model-based reinforcement learning, and

Contribution

In this work we

- learn a high-level representation of actions by embedding the raw action space into a latent action space, and
- learn a world model in the latent space for model-based reinforcement learning, and
- learn a latent policy for planning in theorem proving.

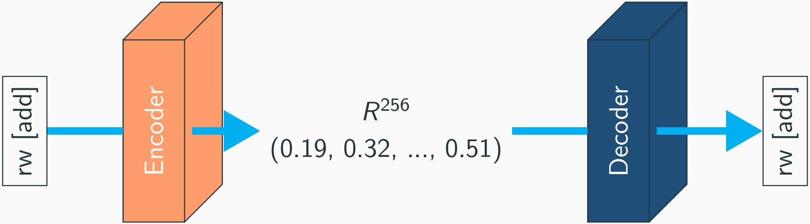
Method

Method — Encoder and Decoder

Learning a high-level representation by embedding the raw action space into a latent action space:

- Encoder_{action}: action space $\rightarrow$ latent action space : $\alpha \sim \texttt{En}(\alpha|a)$.
- Decoder_{action}: latent action space $\rightarrow$ action space : $\hat{a} \sim \texttt{Dn}(\hat{a}|\alpha)$.

Examples — Encoder and Decoder



Method — World Model

Working with the latent action space requires more:

Latent State Space

- State encoder: $z_t \sim \mathtt{En}(z_t|x_t)$
- State decoder: $\hat{x}_t \sim \mathtt{Dn}(\hat{x}_t|z_t)$

Method — World Model

Working with the latent action space requires more:

Latent State Space

- State encoder: $z_t \sim \mathtt{En}(z_t | x_t)$
- State decoder: $\hat{x}_t \sim \mathtt{Dn}(\hat{x}_t | z_t)$

Latent Dynamics

A neural network model to learn the internal dynamics of the theorem proving engine, performing the deduction step of theorem proving.

- Latent transition operator: $\hat{z}_t \sim p(\hat{z}_t | z_{t-1}, \alpha_{t-1})$

Method — Objectives

Reconstruction Loss $\mathcal{L}_{rec}$

```
encoded_action = encoder(raw_actions)
loss = CE(decoder(encoded_actions), raw_actions)
```

Method — Objectives

Reconstruction Loss $\mathcal{L}_{rec}$

```
encoded_action = encoder(raw_actions)
loss = CE(decoder(encoded_actions), raw_actions)
```

Forward Loss $\mathcal{L}_{forward}$

Method — Objectives

Reconstruction Loss $\mathcal{L}_{rec}$

```
encoded_action = encoder(raw_actions)
loss = CE(decoder(encoded_actions), raw_actions)
```

Forward Loss $\mathcal{L}_{forward}$

- $\mathcal{L}_{CE}$: the cross-entropy loss between the ground truth next state and the decoded predicted latent state

Method — Objectives

Reconstruction Loss $\mathcal{L}_{rec}$

```
encoded_action = encoder(raw_actions)
loss = CE(decoder(encoded_actions), raw_actions)
```

Forward Loss $\mathcal{L}_{forward}$

- $\mathcal{L}_{CE}$: the cross-entropy loss between the ground truth next state and the decoded predicted latent state
- $\mathcal{L}_{MSE}$: the mean squared error between the encoded ground truth of next state and the predicted latent state

Method — Objectives

Forward Loss $\mathcal{L}_{forward}$

```
encoded_state = encoder(state)
encoded_action = encoder(action)
predicted_encoded = trans_op(encoded_state, encoded_action)

if using_CE:
    predicted_next_state = decoder(predicted_encoded)
    loss = CE(predicted_next_state, next_state)

elif using_MSE:
    encoded_next_state = encoder(next_state)
    loss = (predicted_encoded - encoded_next_state)^2
```

Method — Objectives

Adding more semantic groundings for the latent state and action space:

$$\mathcal{L} = \mathcal{L}_{rec}(a) + \mathcal{L}_{rec}(x) + \mathcal{L}_{forward}.$$

The latent transition operator allows us to perform efficient planning in the latent space — looking ahead by unrolling the state dynamics for a number of steps.

Method — Latent Policy

Fix the dynamics. From a given latent state, predict a latent action:

- Latent policy: $\alpha_t \sim p(\alpha_t | z_t)$

Fix the dynamics. From a given latent state, predict a latent action:

- Latent policy: $\alpha_t \sim p(\alpha_t | z_t)$
- $\mathcal{L}_{CE}$: the cross-entropy loss between the ground truth target action and the decoded predicted latent action
- $\mathcal{L}_{MSE}$: the mean squared error between the encoded ground truth of target action and the predicted latent action

Experiments

Experiments

- INequality Theorem proving benchmark (INT)

Experiments

- INequality Theorem proving benchmark (INT)
- 40000 proof trajectories
  - cardinality of an axiom combination $K = 3$
  - length of a proof $L = 7$

Experiments

- INequality Theorem proving benchmark (INT)
- 40000 proof trajectories
    - cardinality of an axiom combination $K = 3$
    - length of a proof $L = 7$
- 149009 distinct transitions

Experiments

- INequality Theorem proving benchmark (INT)
- 40000 proof trajectories
  - cardinality of an axiom combination $K = 3$
  - length of a proof $L = 7$
- 149009 distinct transitions
- A character-level transformer for the latent representations of both state and action

Experiments

- INequality Theorem proving benchmark (INT)
- 40000 proof trajectories
  - cardinality of an axiom combination $K = 3$
  - length of a proof $L = 7$
- 149009 distinct transitions
- A character-level transformer for the latent representations of both state and action
- An MLP for internal dynamics (*i.e.*, the transition operator) of INT

Experiments — the INT Environment

INT is a simple and lightweight inequality theorem prover suitable for prototyping our approach. An example proof trajectory in INT:

```
  [{
  "state": "to ((b+a)*(a+b))=(((b+a)*(b+a))*1)",
  "action": "@G((b+a)*(a+b))=(((b+a)*(b+a))*~1)$",
  "next_state": "to ((b+a)*(a+b))=((b+a)*(b+a))"
}, {
  "state": "to ((b+a)*(a+b))=((b+a)*(b+a))",
  "action": "@E((b+a)*~(a+b))=((b+a)*(b+a))$",
  "next_state": "to ((a+b)*(b+a))=((b+a)*(b+a))"
}, {
  "state": "to ((a+b)*(b+a))=((b+a)*(b+a))",
  "action": "@A((a+b)*(b+a))=((b+~a)*(b+a))$",
  "next_state": "QED"
}]
```

Experiments — Transformer Configurations

- 256 embedding dimensions
- 8 attention heads
- 1024 hidden dimensions for position-wise feed-forward layers
- a maximum 128 tokens for both training and evaluation examples.

Metrics:

- How good are the encoders and decoders?

Metrics:

- How good are the encoders and decoders?
  - BLEU scores of reconstructed states and actions

Experiments — Quality of Latent Space

Metrics:

- How good are the encoders and decoders?
    - BLEU scores of reconstructed states and actions
- How good is the transition operator?

Experiments — Quality of Latent Space

Metrics:

- How good are the encoders and decoders?
  - BLEU scores of reconstructed states and actions
- How good is the transition operator?
  - BLEU scores of (decoded) states predicted by the transition operator in one step

Experiments — Quality of Latent Space

Metrics:

- How good are the encoders and decoders?
  - BLEU scores of reconstructed states and actions
- How good is the transition operator?
  - BLEU scores of (decoded) states predicted by the transition operator in one step
  - The change of BLEU scores when we look ahead for more steps

Experiments — Quality of Latent Space

Metrics:

- How good are the encoders and decoders?
  - BLEU scores of reconstructed states and actions
- How good is the transition operator?
  - BLEU scores of (decoded) states predicted by the transition operator in one step
  - The change of BLEU scores when we look ahead for more steps
- How good is the latent policy?

Experiments — Quality of Latent Space

Metrics:

- How good are the encoders and decoders?
  - BLEU scores of reconstructed states and actions
- How good is the transition operator?
  - BLEU scores of (decoded) states predicted by the transition operator in one step
  - The change of BLEU scores when we look ahead for more steps
- How good is the latent policy?
  - BLEU scores of predicted actions and resulting states predicted by the transition operator in one step

Experiments — Quality of Latent Space

Metrics:

- How good are the encoders and decoders?
  - BLEU scores of reconstructed states and actions
- How good is the transition operator?
  - BLEU scores of (decoded) states predicted by the transition operator in one step
  - The change of BLEU scores when we look ahead for more steps
- How good is the latent policy?
  - BLEU scores of predicted actions and resulting states predicted by the transition operator in one step
- Accuracy of predicting "QED"s

Experiments

Table 1: Performance on the test set. $\texttt{BLEU}_{rec-action}$ (resp. $\texttt{BLEU}_{rec-state}$) denotes the BLEU score of the reconstructed actions. $\texttt{BLEU}_{trans}$ denotes the BLEU score of the predicted states by applying the transition operator once. QED accuracy is the percentage of correctly predicted QEDs by applying the transition operator once.

Methods	$\texttt{BLEU}_{rec-action}$	$\texttt{BLEU}_{rec-state}$	$\texttt{BLUE}_{trans}$	QED accuracy (%)
$\mathcal{L}_{CE}$	96.98	94.12	88.23	94.20
$\mathcal{L}_{MSE}$	73.87	69.38	60.18	0

Experiments — Looking ahead

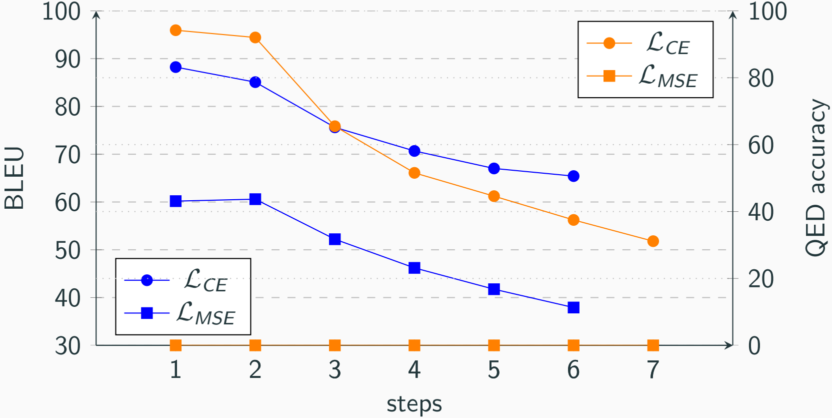


Figure 1: Given a state s, we look ahead n steps by recursively applying the transition operator to s and the subsequent ground truth actions corresponding to s. Note the different scale on right for QED accuracy. Step 7 has a QED accuracy instead of a BLEU score because all target states at step 7 are QEDs.

Experiments — Latent Policy

Table 2: Fix a transition operator learned with $\mathcal{L}_{CE}$. $\texttt{BLEU}_{action}$ denotes the BLEU score of the predicted actions. $\texttt{BLEU}_{next-state}$ denotes the BLEU score of the predicted next states by applying the transition operator once. QED accuracy is the percentage of correctly predicted QEDs by applying the transition operator once to the state and predicted action.

Methods	$\texttt{BLEU}_{action}$	$\texttt{BLEU}_{next-state}$	QED accuracy (%)
$\mathcal{L}_{CE}$	86.73	76.05	18.60
$\mathcal{L}_{MSE}$	85.88	81.82	79.55