Semi-**S**upervised Policy **I**nitialization for Playing Games with Language Hints

Tsu-Jui Fu

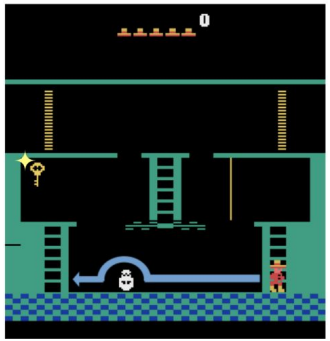
William Wang

UC Santa Barbara

https://tsujuifu.github.io/pubs/naacl21_ssi.pdf

Playing Games with Language Hint

- Most RL methods rely on exploration and maximization the reward
- A common setting is only to give out the **achieved signal**
  - Reward is **1 if achieved**, otherwise 0
  - The **sparse-reward** makes the agent difficult to learn



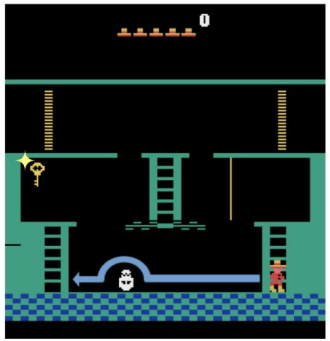
if achieved

final goal

Sparse-Reward

Playing Games with Language Hint

- Most RL methods rely on exploration and maximization the reward
- A common setting is only to give out the **achieved signal**
- **Language hint** can be seen as an additional **dense-reward**
  - If agent's **actions correlate** with the given hint



if achieved → **final goal** : **Sparse**-Reward

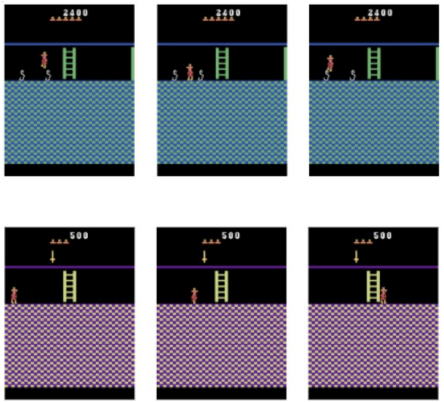
correlation → **language hint** : **Dense**-Reward

Playing Games with Language Hint

- Conducted tasks in the experiments
  - 45 tasks under Atari **Montezuma's Revenge**
  - Each task contains **starting state**, unknown **final goal**, and **a given hint**

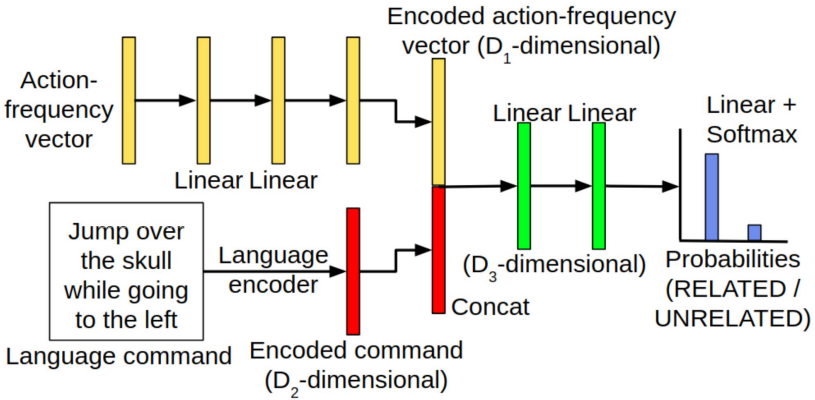
Baseline: ExtLang

- [IJCAI'19] Using Language for Reward Shaping in Reinforcement Learning
- Collect **human demo clips** to learn the correlation between action and hint



jump twice to the left

walk to the right

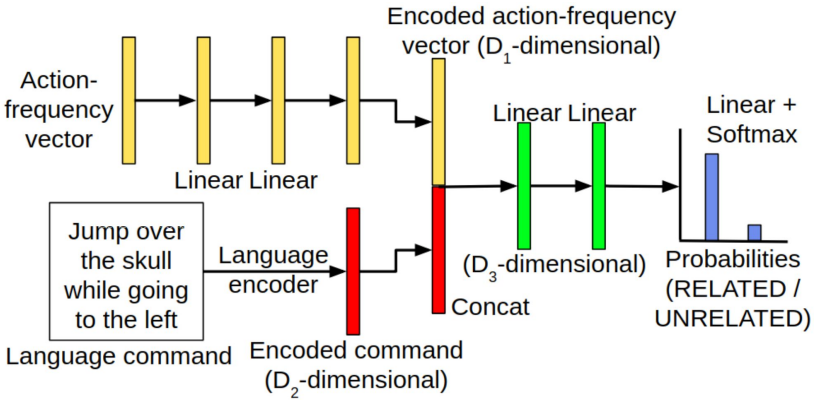


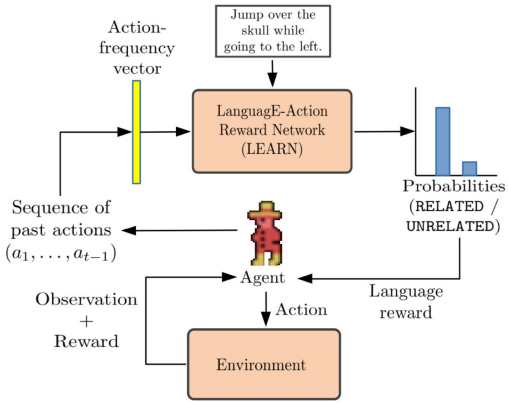
https://www.ijcai.org/Proceedings/2019/331

Baseline: ExtLang

- [IJCAI'19] Using Language for Reward Shaping in Reinforcement Learning

- Collect **human demo clips** to learn the correlation between action and hint
- Apply **reward**$_{env}$ and **reward**$_{hint}$ during task-training
  - **Reward**$_{hint}$ is provided each action ([0~1] as **dense**-reward)

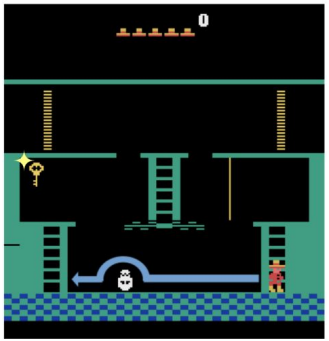




https://www.ijcai.org/Proceedings/2019/331

How about Latent Hints?

- Each task contains **only one** given hint, which is **incomplete**
  - Those **latent hints** rely on exploring the environment
  - Still suffer from **sparse-reward** issue



jump over the skull

climb up the ladder

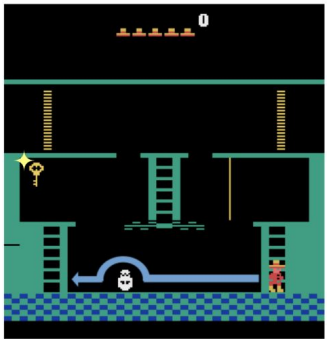
jump to get the key

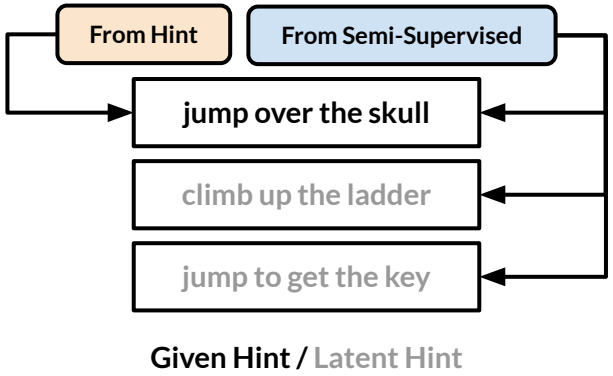
Given Hint / Latent Hint

How about Latent Hints?

- Each task contains **only one** given hint, which is **incomplete**

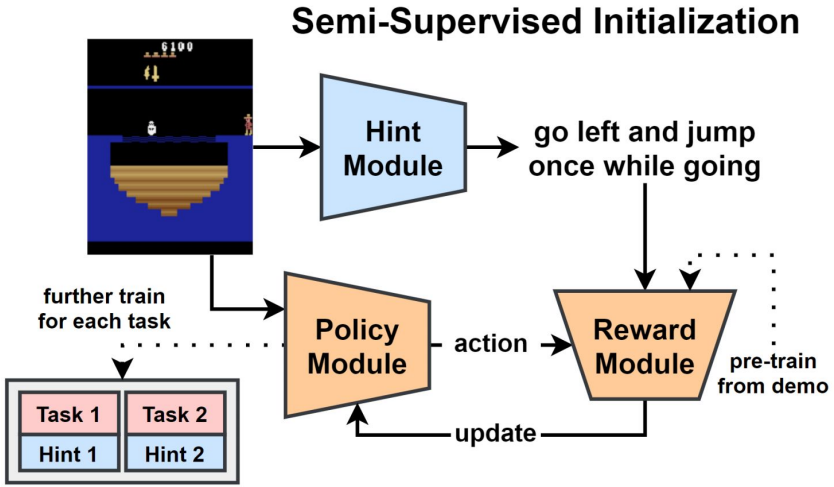
- Semi-Supervised Initialization (SSI)
  - Enable agent to **experience various possible hints** in advance
  - Can have a **better-initialized policy** during task-training





Semi-Supervised Initialization (SSI)

1. Hint module ***H*** generates possible hints ***l*** for random state ***s***
2. Policy module ***P*** rollouts and steps actions ***a***
3. Reward module ***R*** updates ***P*** based on (***a***, ***l***)
4. Different ***s*** to ***H*** let ***P*** learn from different possible ***l***
5. ***P*** serves as better-initialized during further **task-training**

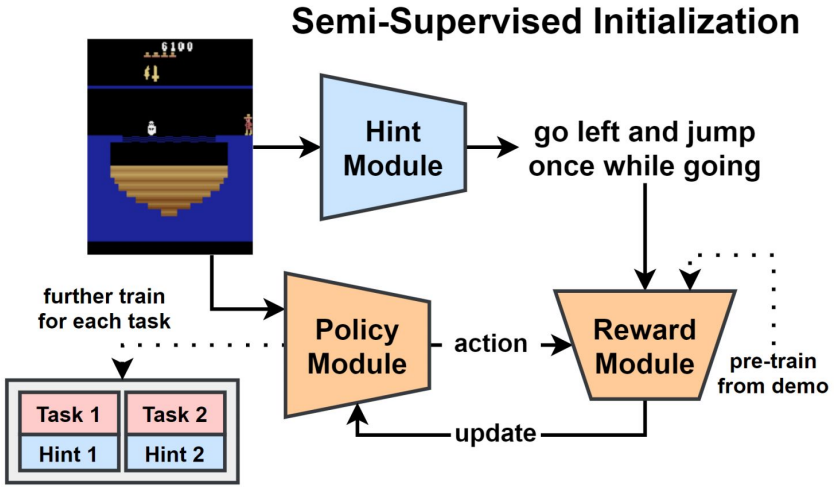


https://tsujuifu.github.io/pubs/naacl21_ssi.pdf

Semi-Supervised Initialization (SSI)

1. Hint module ***H*** generates possible hints ***l*** for random state ***s***
   - Adopt CNN to extract visual feature from ***s*** and GRU to decode ***l***
   - ***H***: ***l*** = GRU(CNN(***s***))

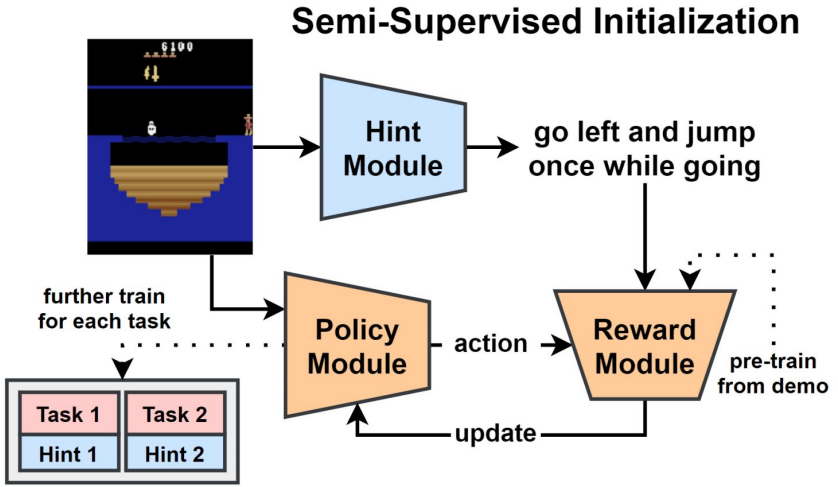
   - ***H*** is pre-trained on collected **human demo clips**



Semi-Supervised Initialization (SSI)

1. Hint module ***H*** generates possible hints ***l*** for random state ***s***

2. Policy module ***P*** rollouts and steps actions ***a***
   - Recurrent action selector to step a_t for s_t
   - ***P***: $a_t = \text{GRU}(s_t \mid h_t)$

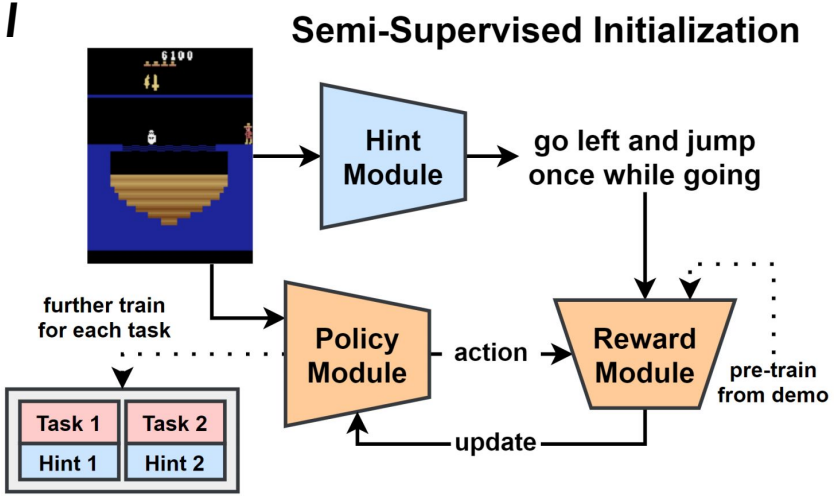


Semi-Supervised Initialization (SSI)

1. Hint module ***H*** generates possible hints ***l*** for random state ***s***

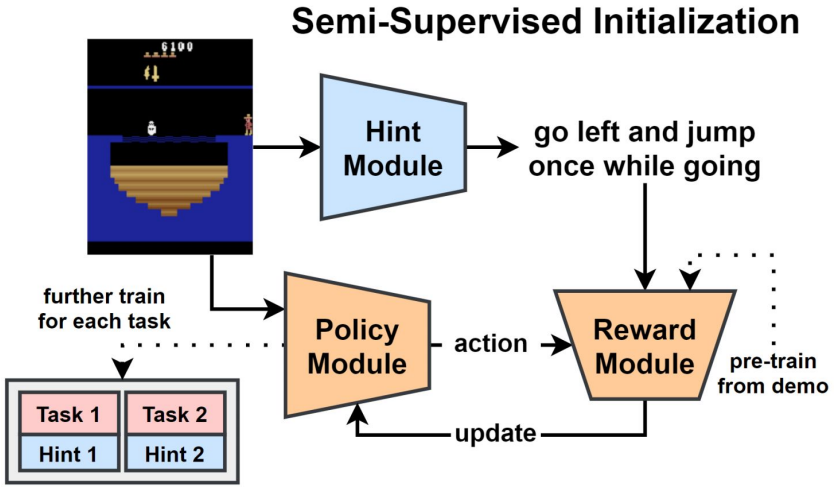
2. Policy module ***P*** rollouts and steps actions ***a***

3. Reward module ***R*** updates ***P*** based on (***a***, ***l***)
   - ***R*** provides correlation between ***a*** and ***l***
   - ***R***: ***r*** ([0~1]) = Binary Classifier(***a***, ***l***)
   - Update ***P*** to maximize ***r*** via PPO



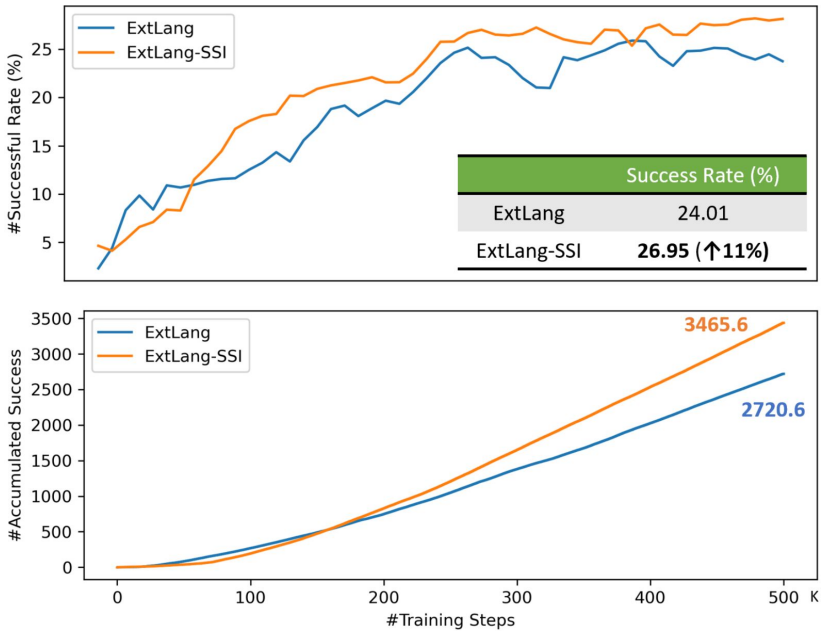
Semi-Supervised Initialization (SSI)

1. Hint module ***H*** generates possible hints ***l*** for random state ***s***
2. Policy module ***P*** rollouts and steps actions ***a***
3. Reward module ***R*** updates ***P*** based on (***a***, ***l***)
4. Different ***s*** to ***H*** let ***P*** learn from different possible ***l***
5. ***P*** serves as better-initialized during further **task-training**
    - Similar to ExtLang, optimize by both **reward**$_{env}$ and **reward**$_{hint}$



Experiments (ExtLang-SSI vs ExtLang)

- Success Rate
  - ExtLang-SSI has **higher suc. rate** under the **same training step**
  - **11% relative improvement** of final policy

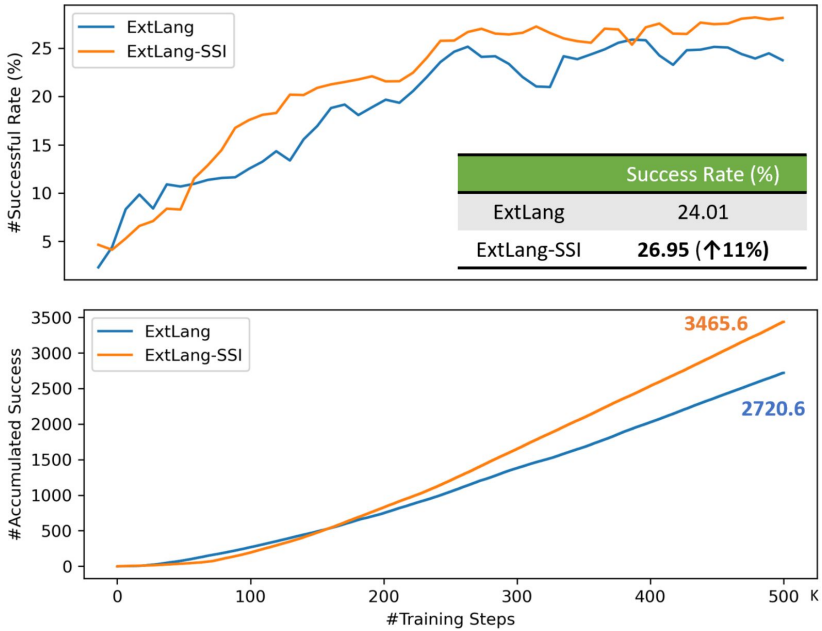


Experiments (ExtLang-SSI vs ExtLang)

- Success Rate

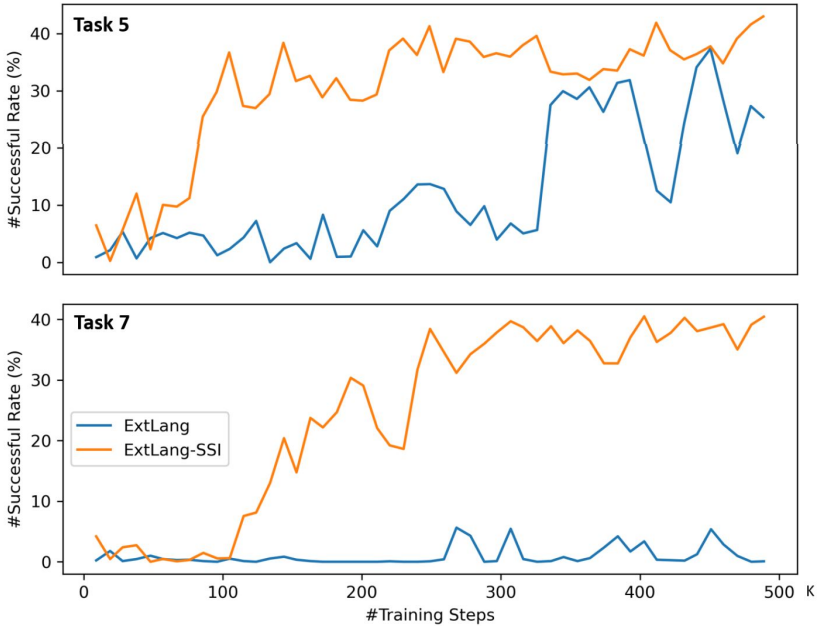
- **Accumulated** Success Episodes
  - ExtLang-SSI learns faster (**420K steps** vs 500K for 2.7K episodes)

  - **1.2x speed up** during task-training



Experiments (ExtLang-SSI vs ExtLang)

- Success Rate
- Accumulated Success Episodes
- Task Study
  - ExtLang has 35% for **task 5** but ExtLang-SSI achieves using only 100K steps
  - ExtLang almost fails for **task 7** but ExtLang-SSI finally reach 40%



Qualitative Results of Hint Module

- Policy module can learn from **latent but useful hints** generated by hint module