Align-RUDDER: Learning From Few Demonstrations by Reward Redistribution

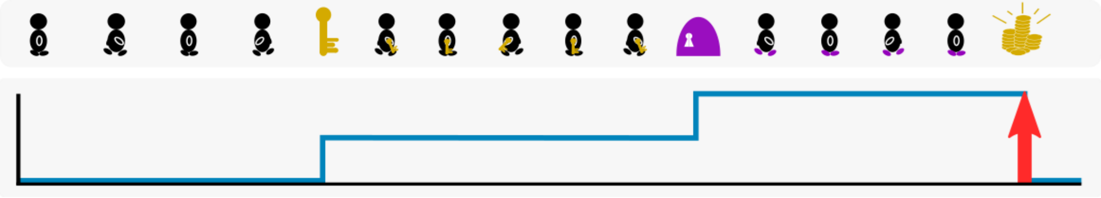
Vihang Patil*, Markus Hofmarcher*, Marius-Constantin Dinu, Matthias Dorfer, Patrick Blies, Johannes Brandstetter, Jose Arjona-Medina, Sepp Hochreiter

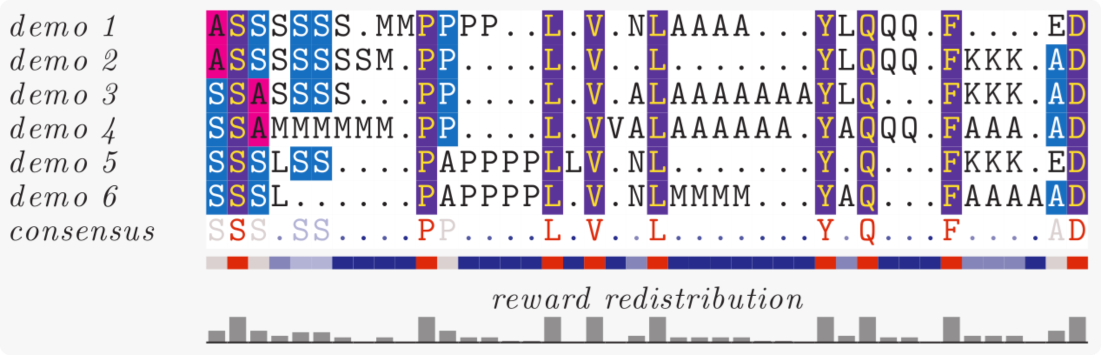
* Equal contribution

JKU JOHANNES KEPLER UNIVERSITY LINZ

enliteAI

Align-RUDDER in a Nutshell

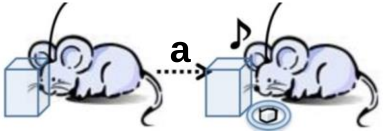




```
demo 1      ASSSSSS.MMPPPP..L.V.NLAAAA...YLQQQ.F....ED
demo 2      ASSSSSSSM.PP....L.V..L.......YLQQQ.FKKK.AD
demo 3      SSASSSS...PP....L.V.ALAAAAAAAYLQ...FKKK.AD
demo 4      SSAMMMMMM.PP....L.VVALAAAAAA.YAQQQ.FAAA.AD
demo 5      SSSLSS....PAPPPPLLV.NL.......Y.Q...FKKK.ED
demo 6      SSSL......PAPPPPL.V.NLMMMM...YAQ...FAAAAAD
consensus   SSS.SS....PP....L.V..L.......Y.Q...F....AD
```

reward redistribution

Complex Tasks have Delayed Rewards



Complex tasks often have episodic rewards:

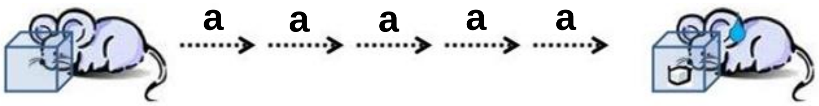
- Actions cause reward or penalty that is **obtained much later**
- **Distracting rewards** may be present
- **Credit assignment problem**: which action was responsible?

CHAPTER I

INTRODUCTION

The credit-assignment problem for a complex learning system (Minsky, 1961) is the problem of properly assigning credit or blame for overall outcomes to each of the learning system's internal decisions that contributed to those outcomes. In many

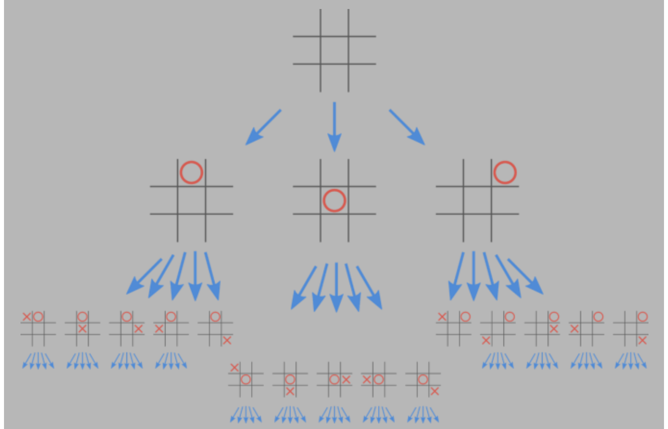
[1] R. S. Sutton, ‘Temporal Credit Assignment in Reinforcement Learning’, University of Massachusetts, 1984



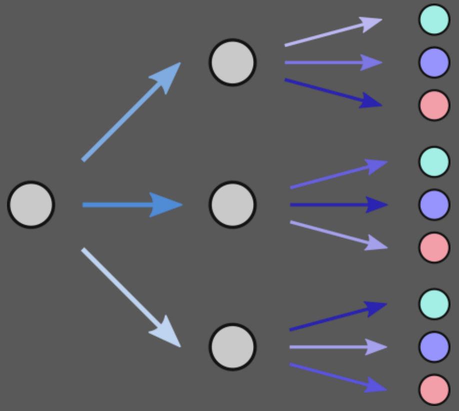
The Problem of TD and MC

- Traditional approaches make guesses about the future
- Correcting the *bias of temporal difference* (TD) learning (SARSA and Q-learning) requires **exponential updates**
- Monte Carlo (MC) methods have **high variance** since *variance is propagated through all states* that are visited

High Branching Factor

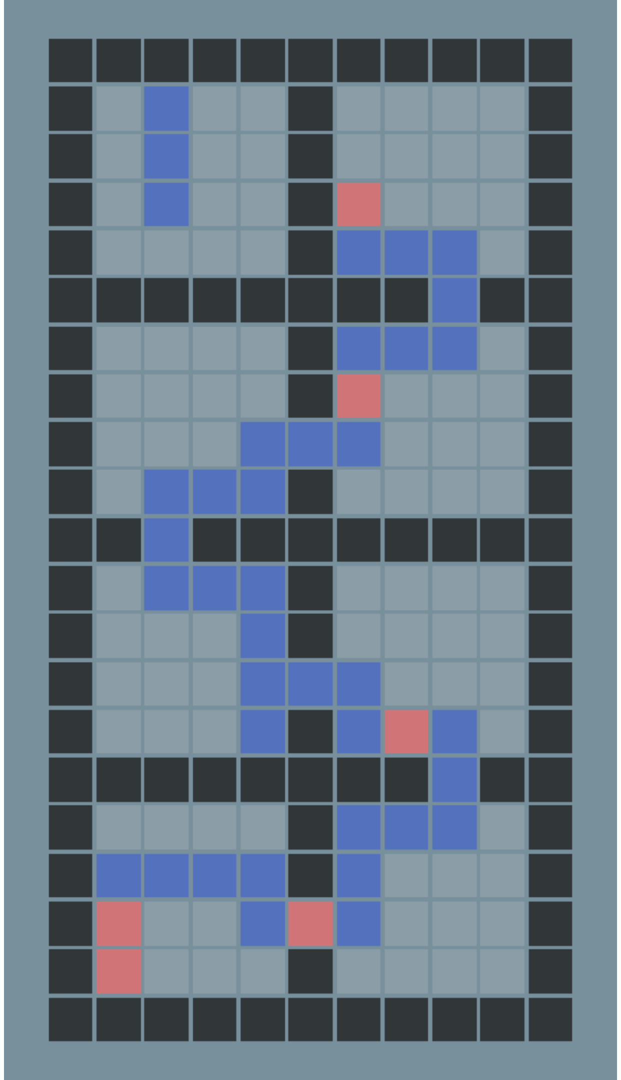


Probabilistic Transitions



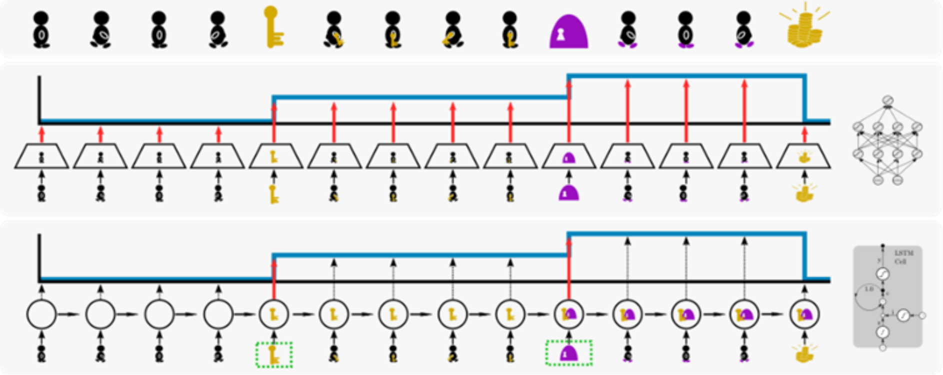
Detecting Key Events

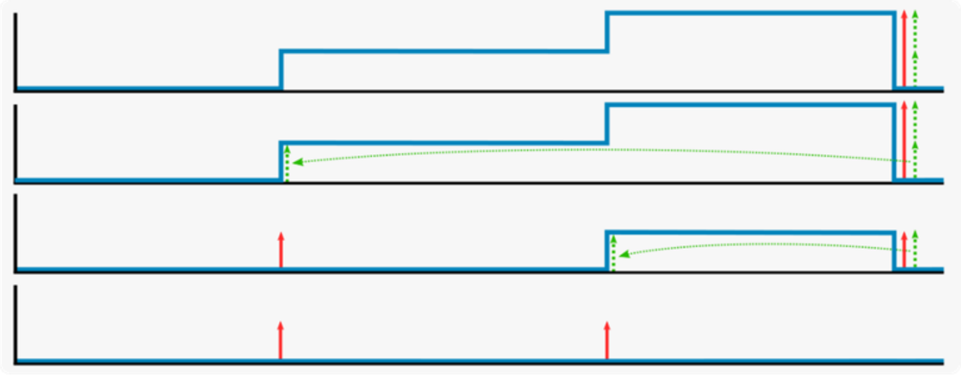
- Analyze episodes that have been observed
  - No probabilities and **no guesses about the future**
  - Detect **key events** that lead to rewards (i.e. sub-goals)
- **Supervised** learning problem
- Example: RUDDER [2]



RUDDER: Reward Redistribution to Key Events

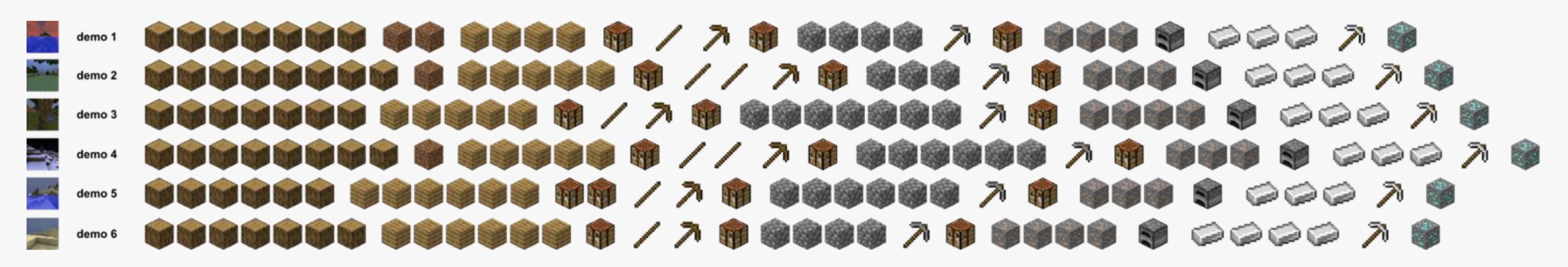
- Give immediate feedback
- Reward is the difference in the expected return (RUDDER [2])
- Reduces the delay of rewards
- Identifies key events and landmarks





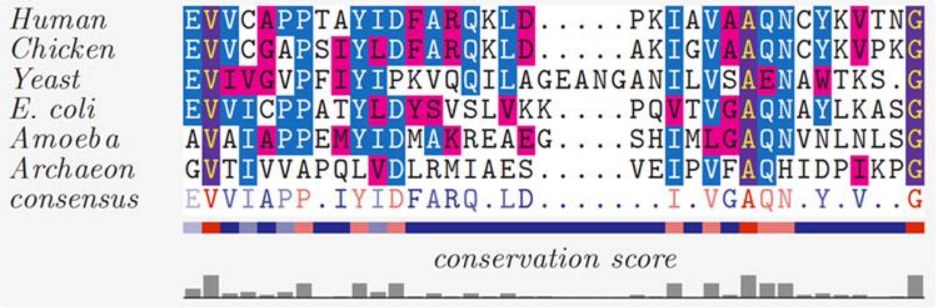
Few Demonstrations

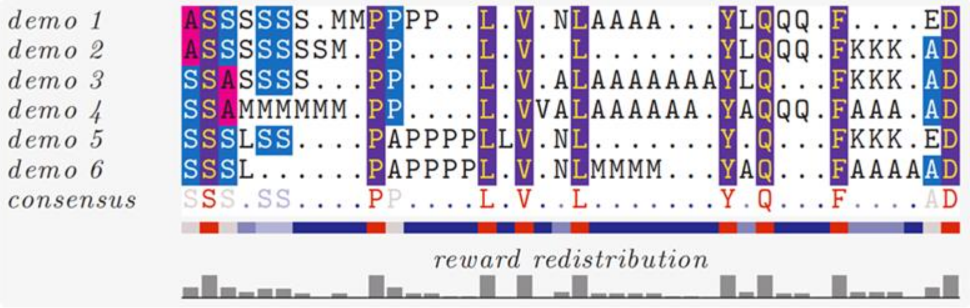
- Often only few **expert demonstrations** available
- Training an LSTM model…
  - …is difficult from a **small number of demonstrations**
  - …requires high and low return examples



Sequence Alignment for Reward Redistribution

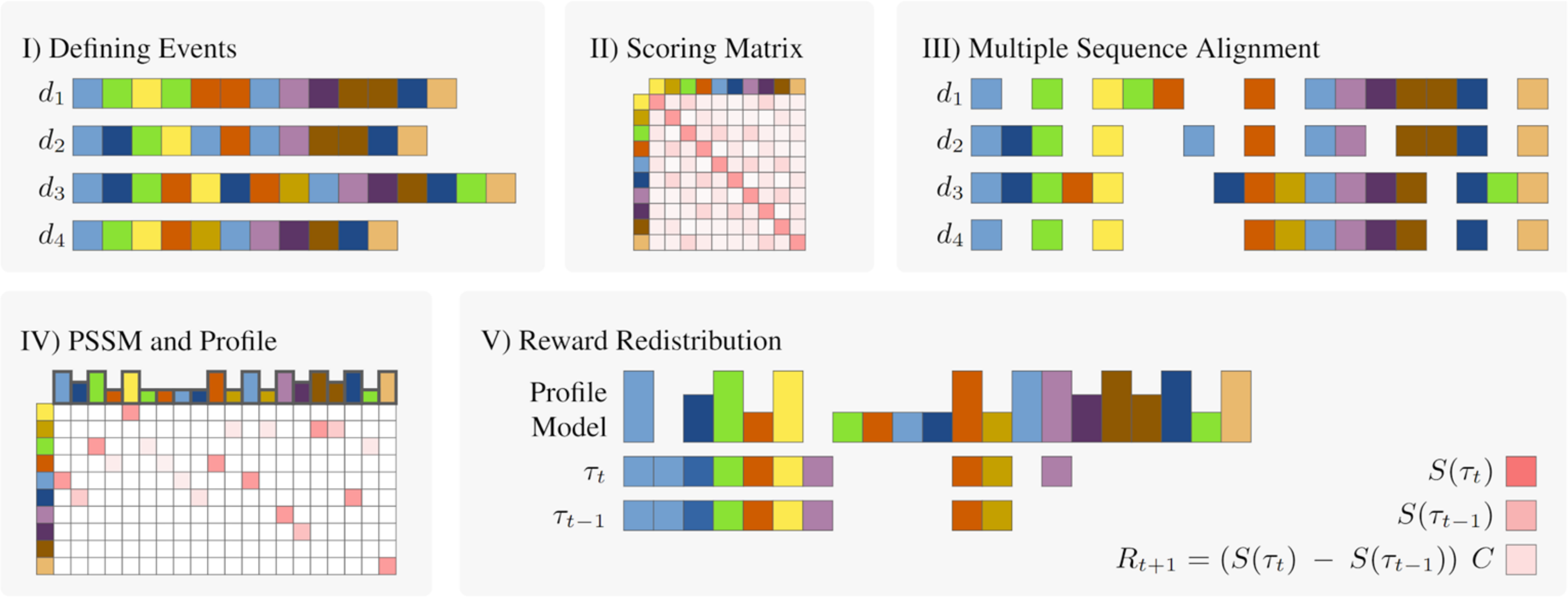
- Sequence alignment works with a **small number of examples**
- Sequence alignment uses only closely related examples
- The result of such an alignment is a profile model
- New sequences are aligned to a profile model and receive an **alignment score**
- The redistributed reward is proportional to the difference of scores of consecutive time steps


Human EVVCAPPTAYIDFARQKLD.....PKIAVAAQNCYKVTNG
Chicken EVVCGAPSIYLDFARQKLD.....AKIGVAAQNCYKVPKG
Yeast EVIVGVPFIYIPKVQQILAGEANGANILVSAENAWTKS.G
E. coli EVVICPPATYLDYSVSLVKK....PQVTVGAQNAYLKASG
Amoeba AVAIAPPEMYIDMAKREAEG....SHIMLGAQNVNLNLSG
Archaeon GVTIVVAPQLVDLRMIAES.....VEIPVFAQHIDPIKPG
consensus EVVIAPP.IYIDFARQ.LD.......I.VGAQN.Y.V..G
conservation score


demo 1 ASSSSSS.MMPPPP..L.V.NLAAAA...YLQQQ.F....ED
demo 2 ASSSSSSSM.PP....L.V..L........YLQQQ.FKKK.AD
demo 3 SSASSSS...PP....L.V.ALAAAAAAAYLQ...FKKK.AD
demo 4 SSAMMMMMM.PP....L.VVALAAAAAAA.YAQQQ.FAAA.AD
demo 5 SSSLSS....PAPPPPLLV.NL........Y.Q...FKKK.ED
demo 6 SSSL......PAPPPPL.V.NLMMMM...YAQ...FAAAAAD
consensus SSS.SS....PP....L.V..L........Y.Q...F....AD
reward redistribution

Align-RUDDER

Align-RUDDER



I) Defining Events

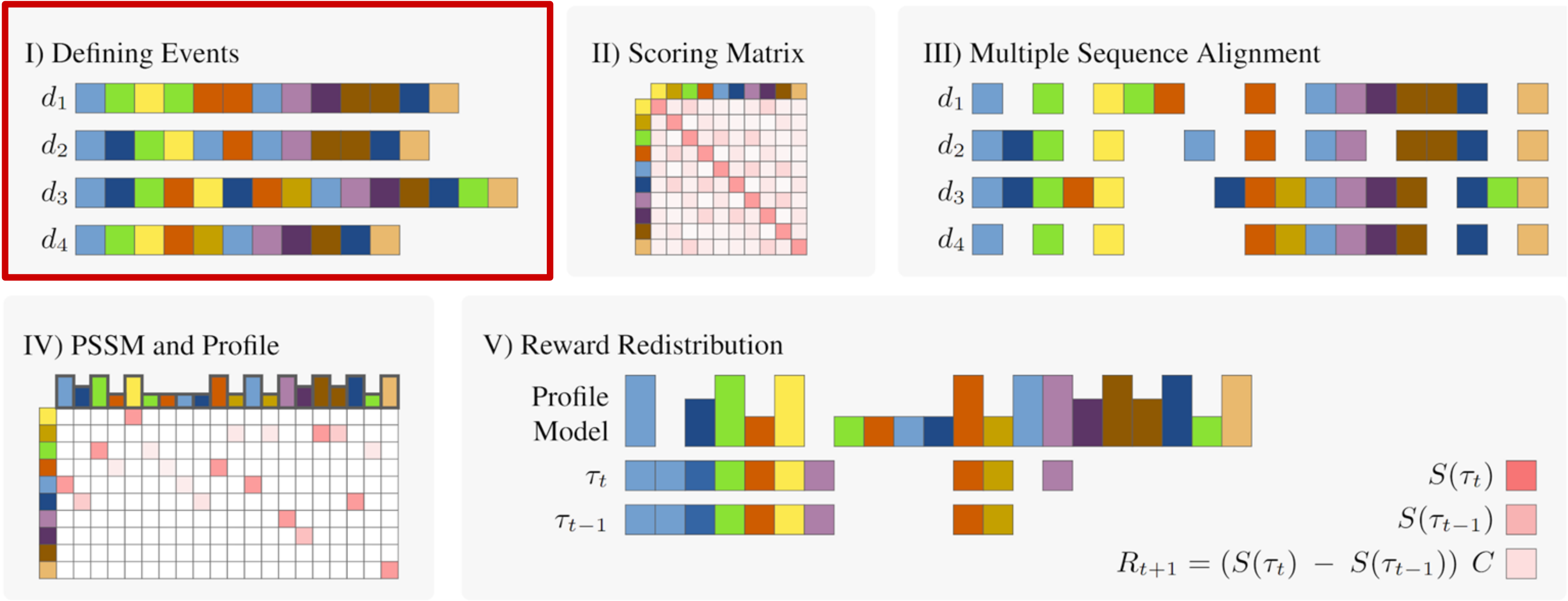
II) Scoring Matrix

III) Multiple Sequence Alignment

IV) PSSM and Profile

V) Reward Redistribution

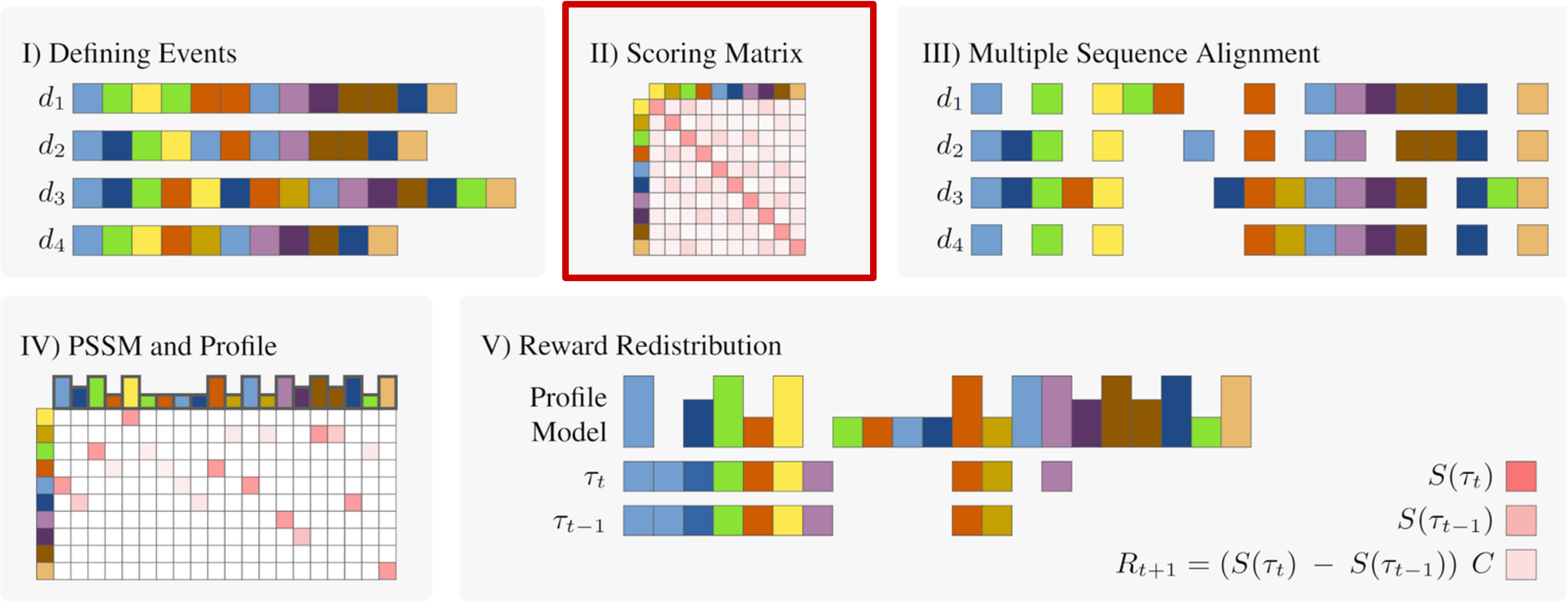
Align-RUDDER



JOHANNES KEPLER UNIVERSITY LINZ

enliteAI

Align-RUDDER



I) Defining Events

II) Scoring Matrix

III) Multiple Sequence Alignment

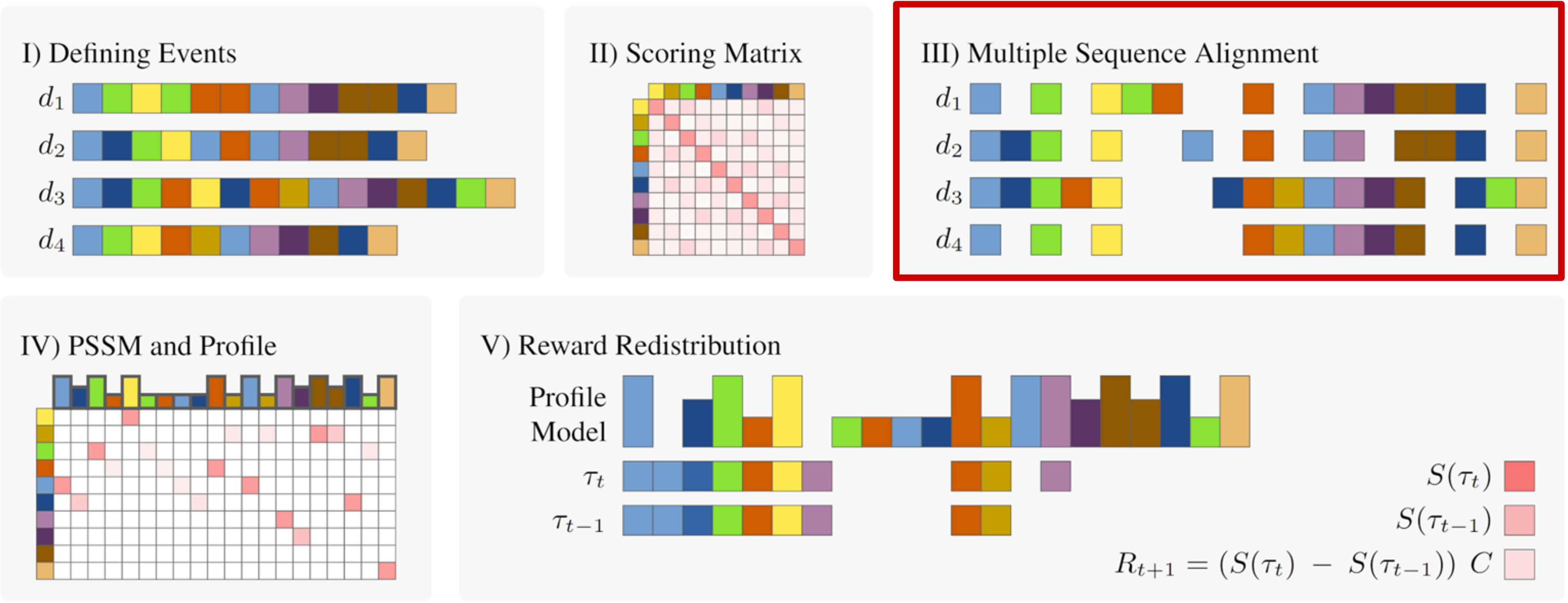
IV) PSSM and Profile

V) Reward Redistribution

JKU JOHANNES KEPLER UNIVERSITY LINZ

enliteAI

Align-RUDDER



I) Defining Events

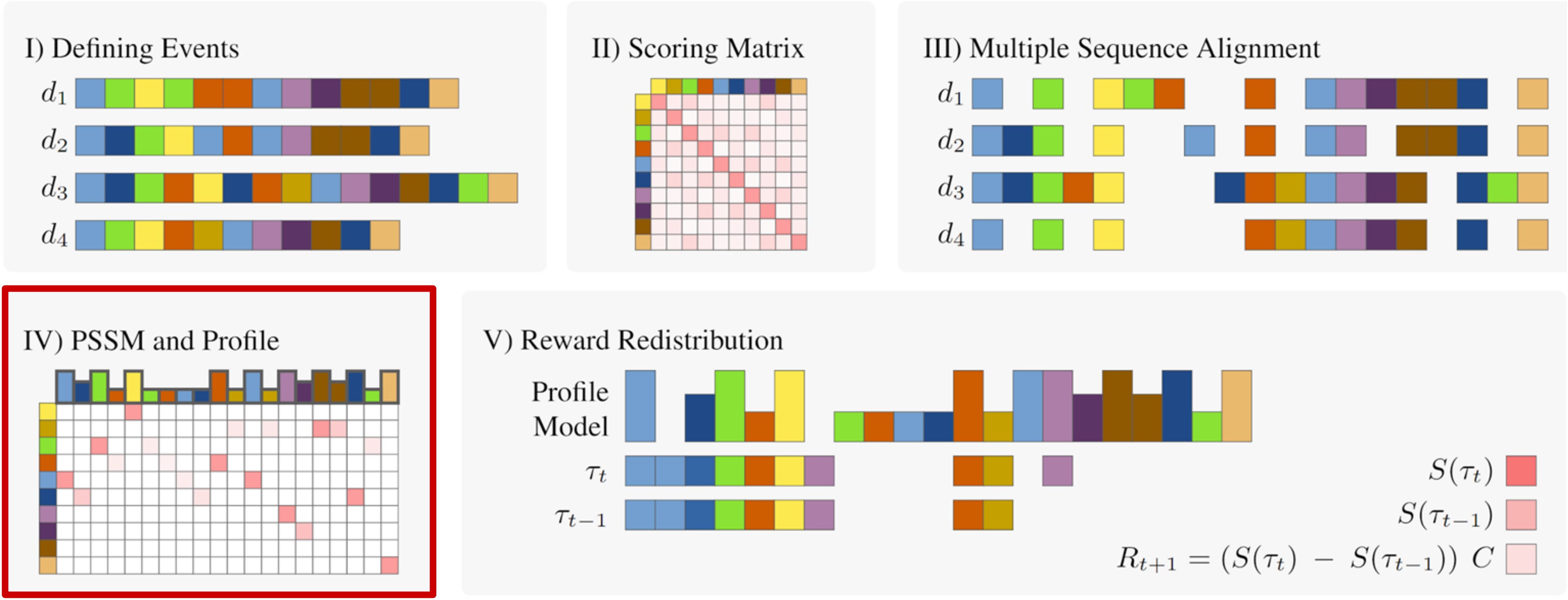
II) Scoring Matrix

III) Multiple Sequence Alignment

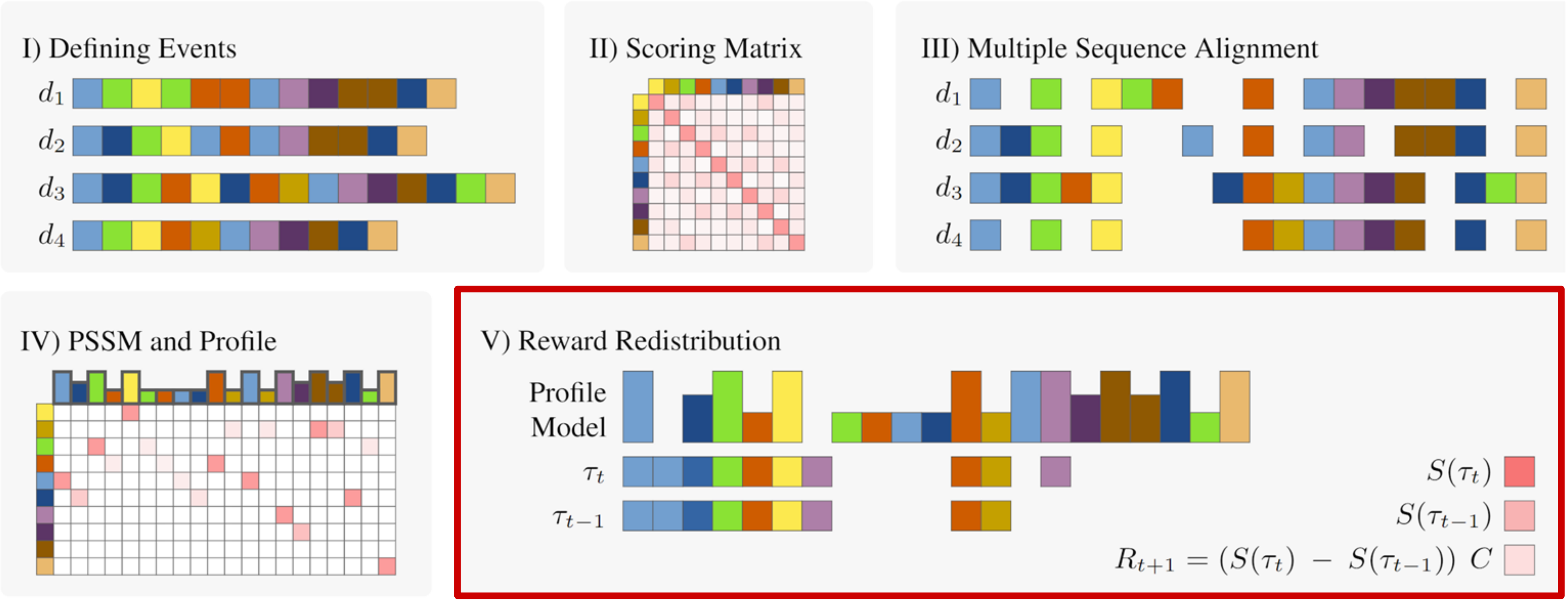
IV) PSSM and Profile

V) Reward Redistribution

Align-RUDDER



Align-RUDDER



I) Defining Events

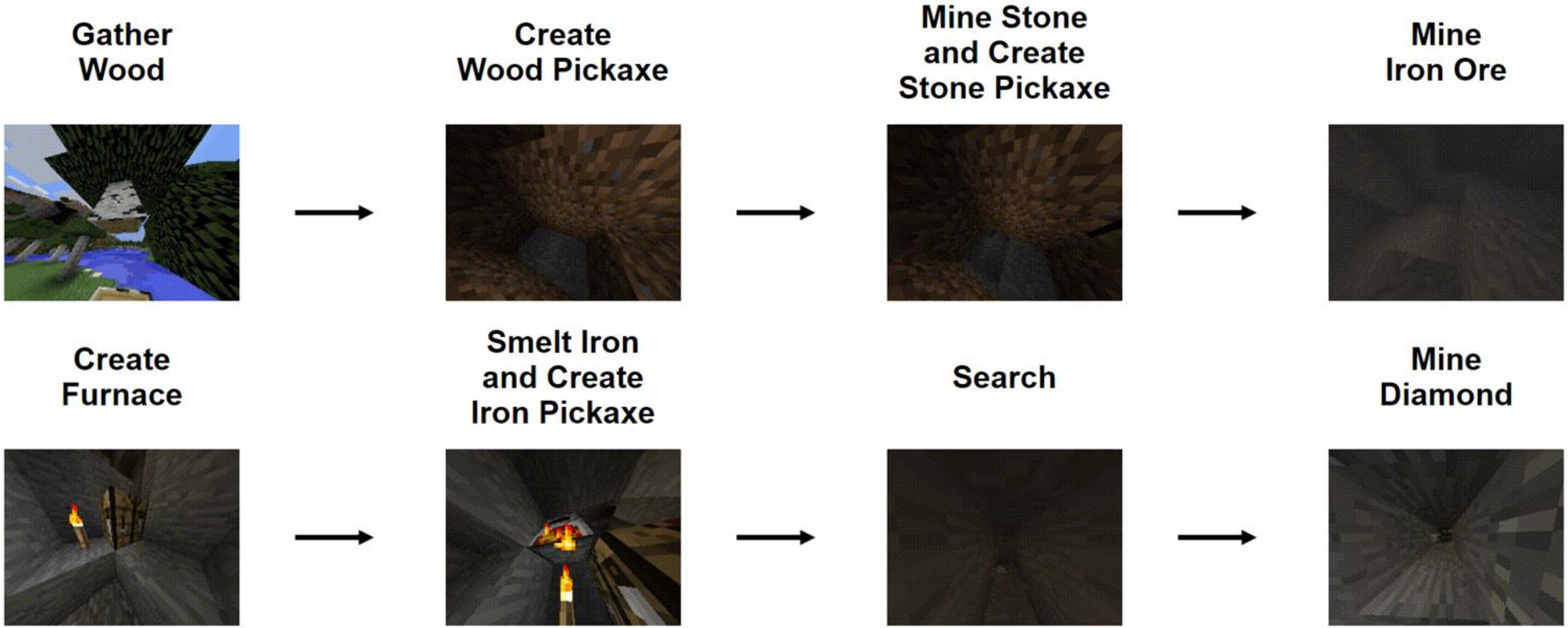
II) Scoring Matrix

III) Multiple Sequence Alignment

IV) PSSM and Profile

V) Reward Redistribution

Mining a Diamond in Minecraft

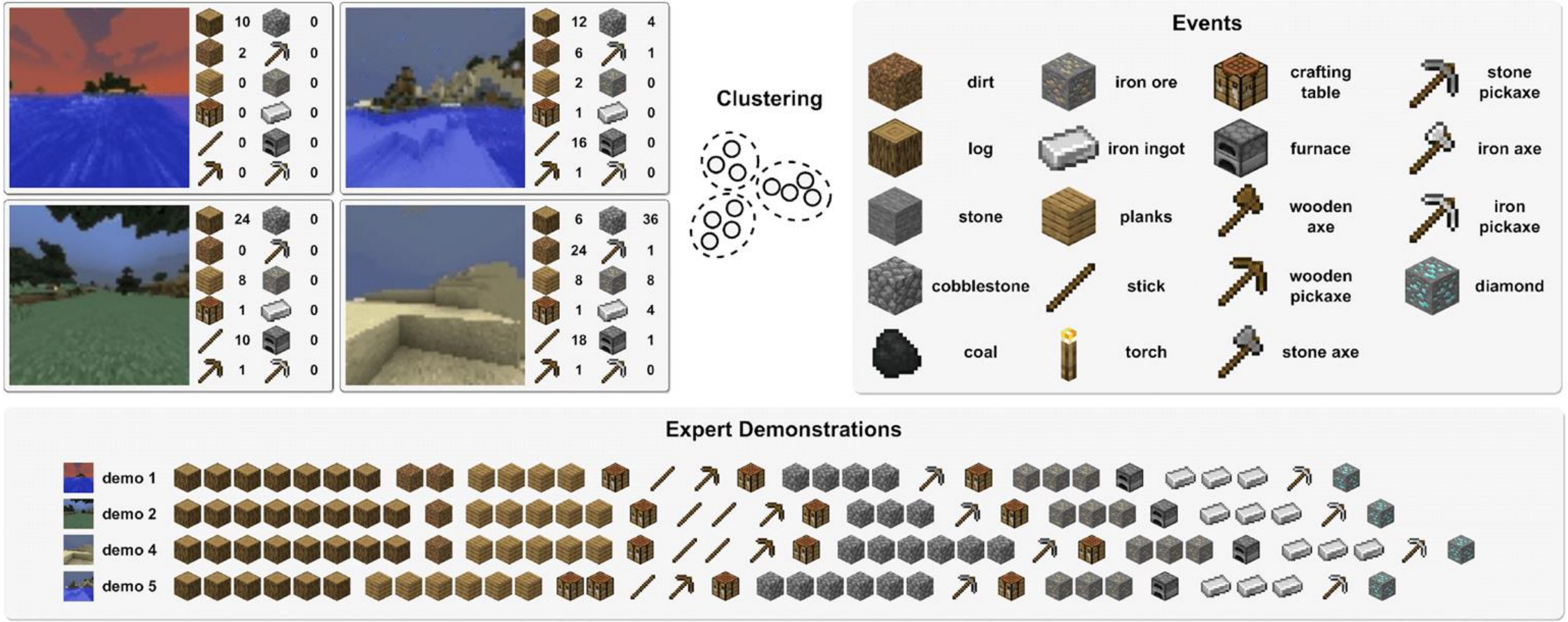


Gather Wood → Create Wood Pickaxe → Mine Stone and Create Stone Pickaxe → Mine Iron Ore

Create Furnace → Smelt Iron and Create Iron Pickaxe → Search → Mine Diamond

Image taken from MineRL [3]

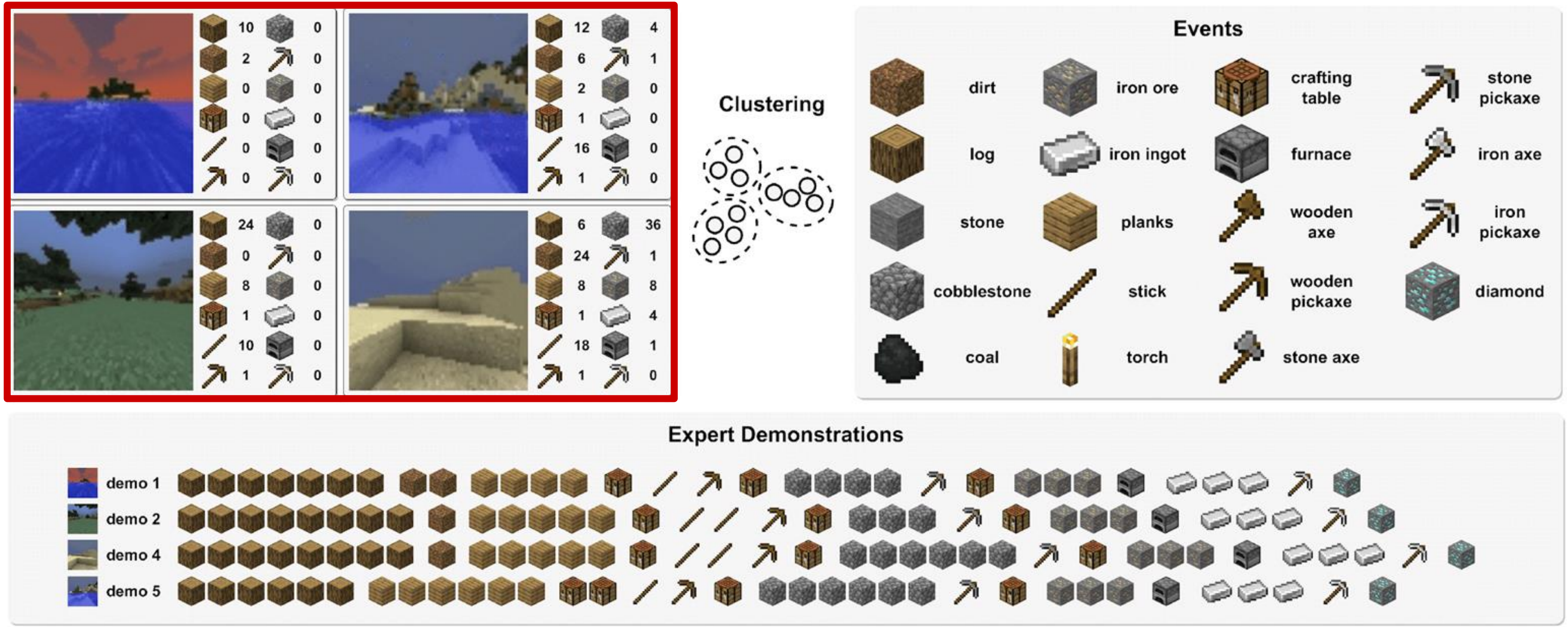
(I) Define Events



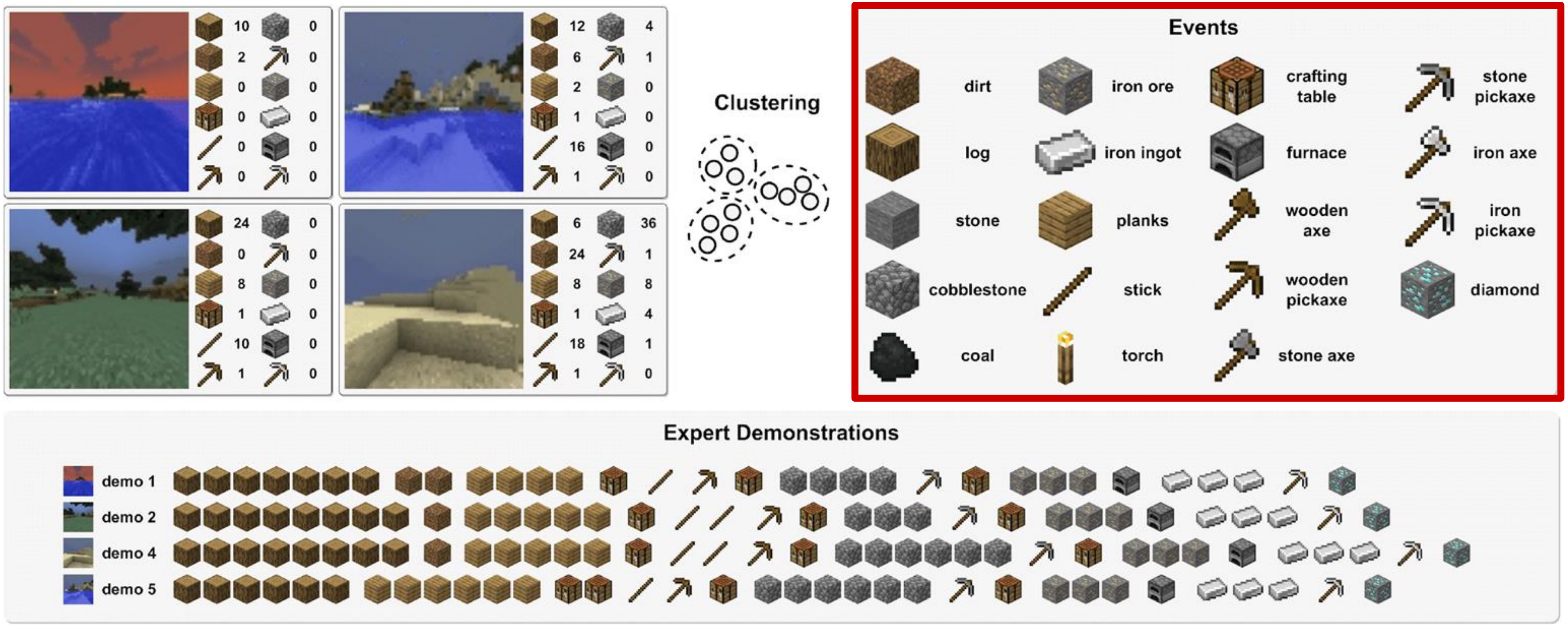
JOHANNES KEPLER
UNIVERSITY LINZ

enliteAI

(I) Define Events



(I) Define Events



Events

dirt, iron ore, crafting table, stone pickaxe

log, iron ingot, furnace, iron axe

stone, planks, wooden axe, iron pickaxe

cobblestone, stick, wooden pickaxe, diamond

coal, torch, stone axe

Expert Demonstrations

demo 1

demo 2

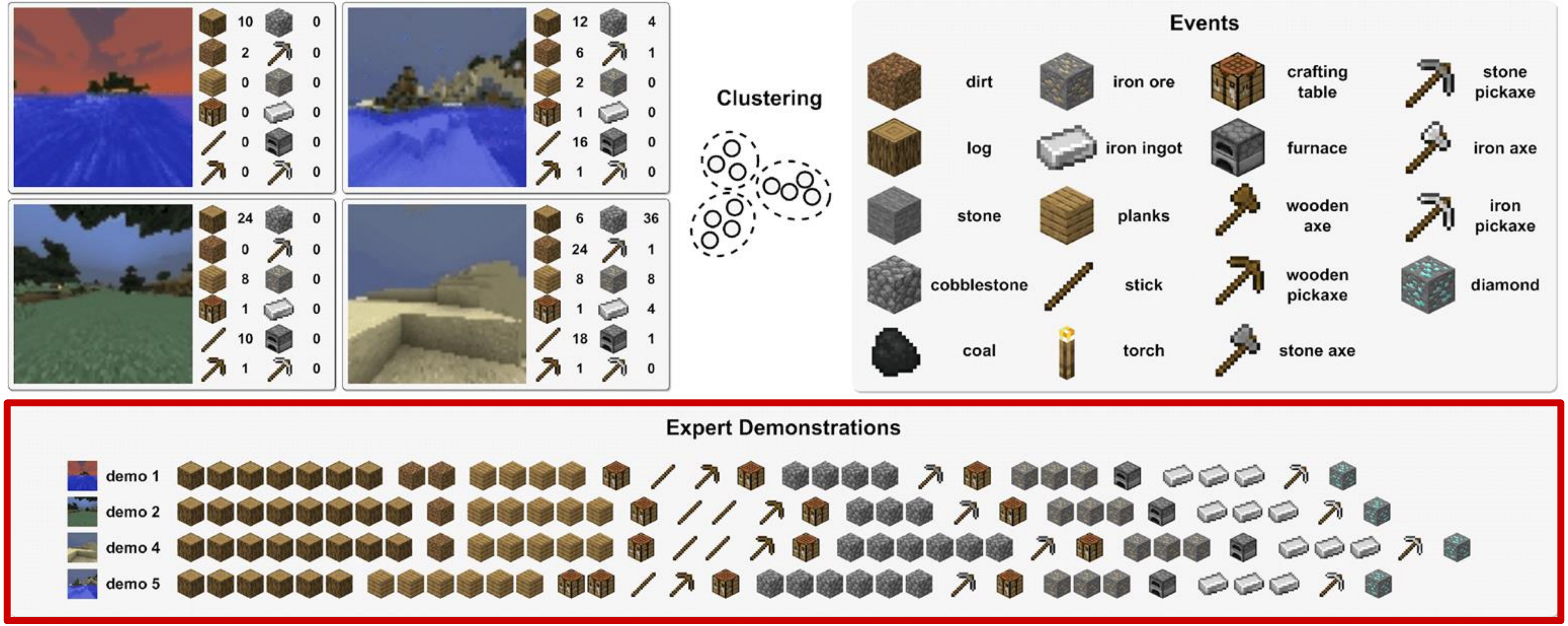
demo 4

demo 5

JOHANNES KEPLER
UNIVERSITY LINZ

enliteAI

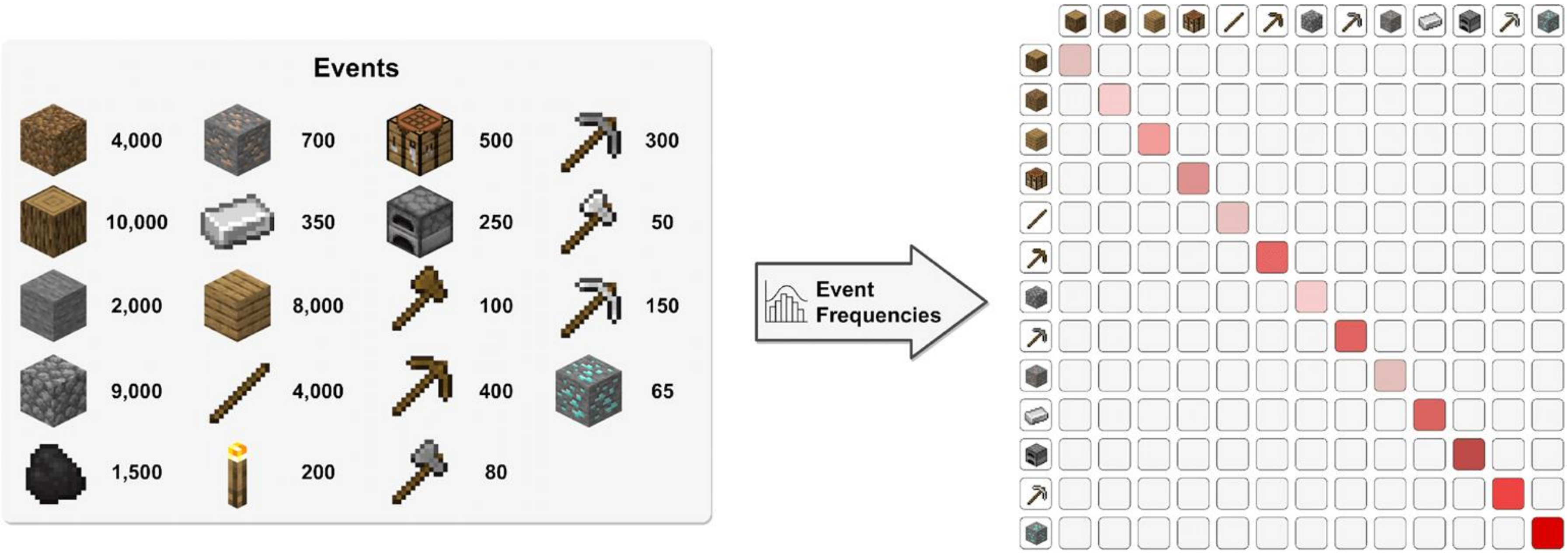
(I) Define Events



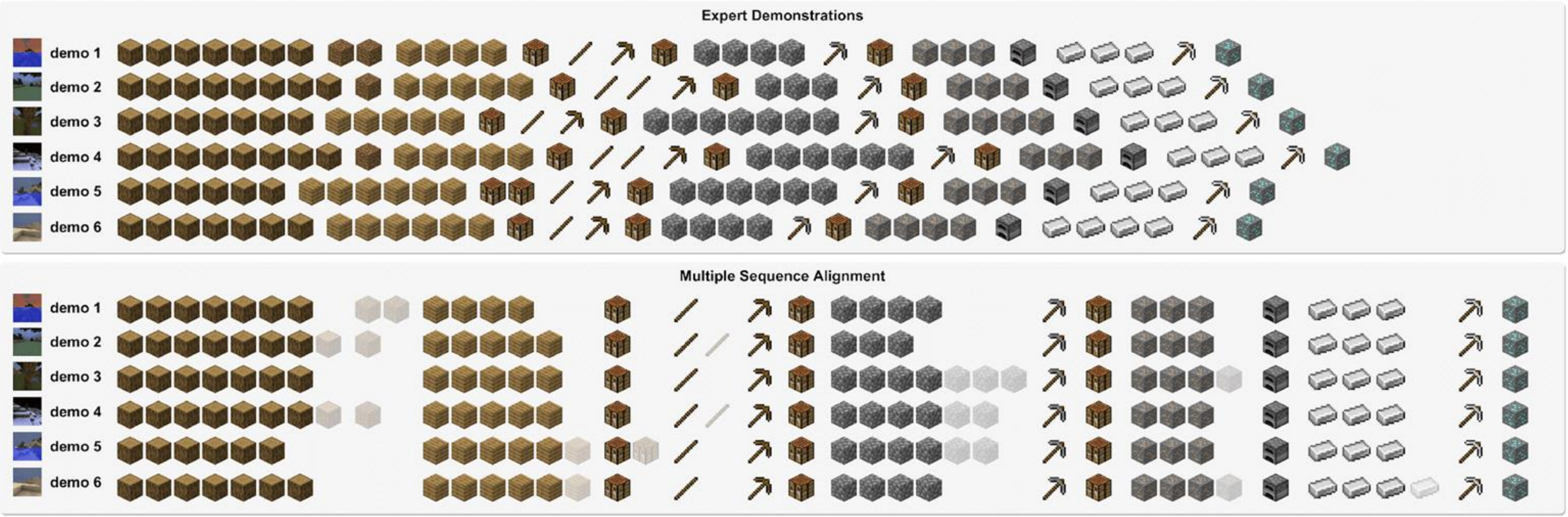
JOHANNES KEPLER UNIVERSITY LINZ

enliteAI

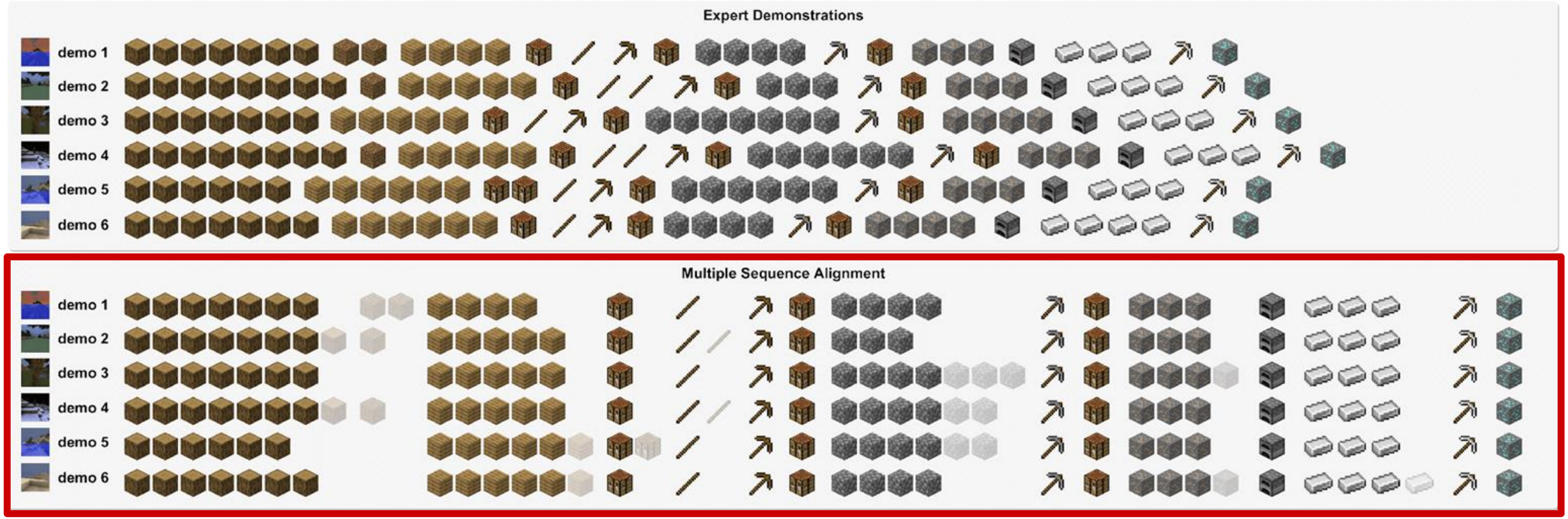
(II) Determine the Scoring Matrix



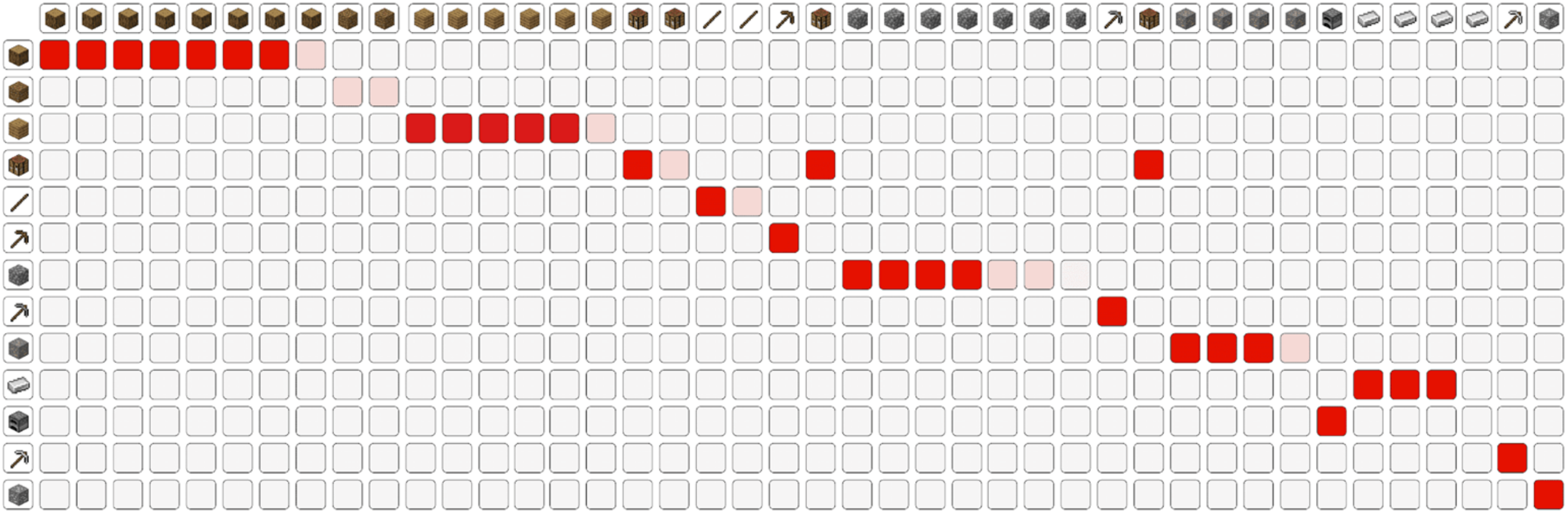
(III) Multiple sequence alignment (MSA)



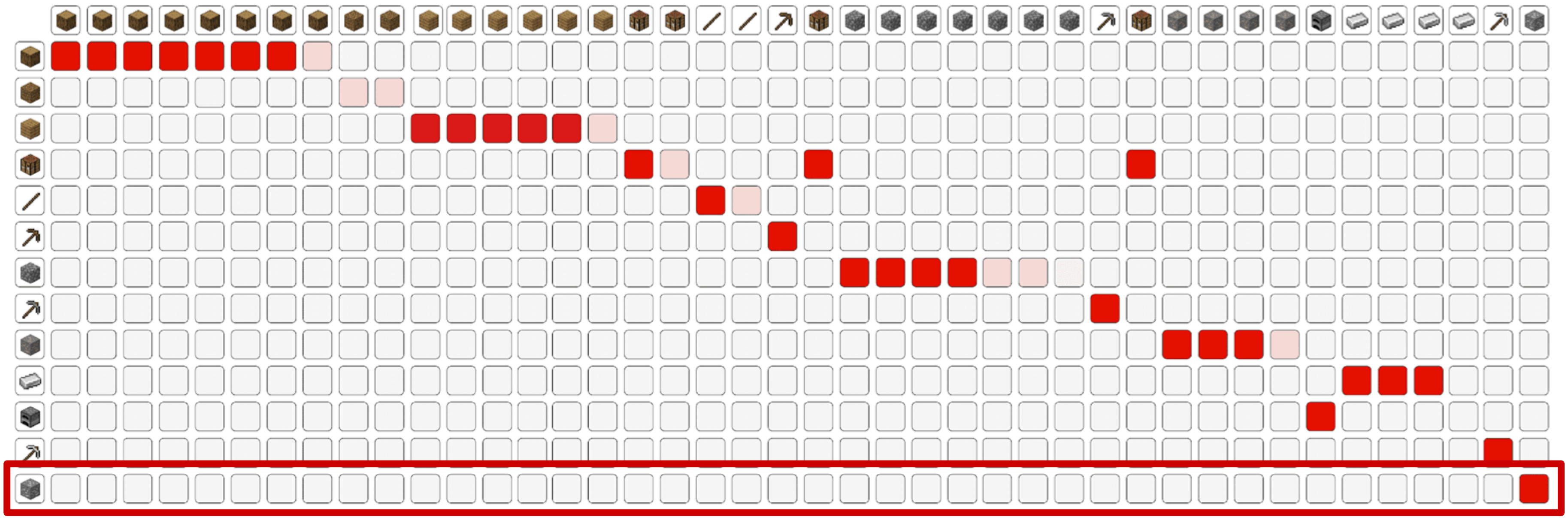
(III) Multiple sequence alignment (MSA)



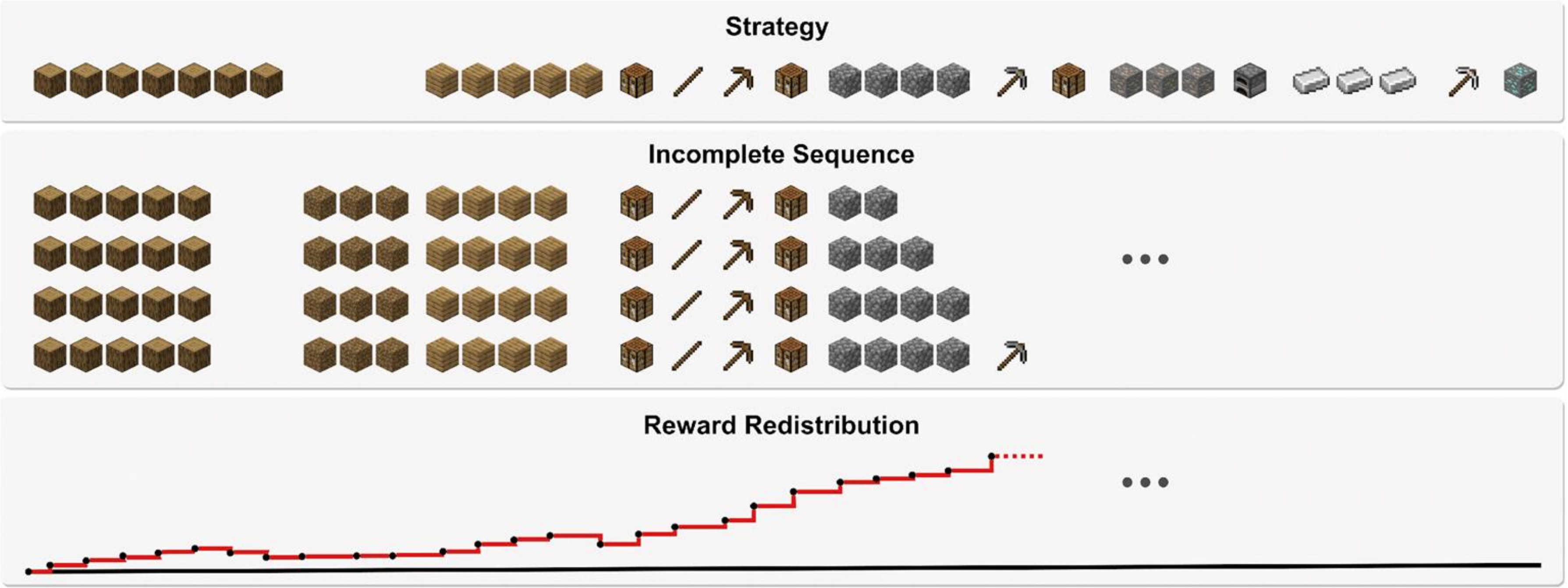
(IV) Position-specific Scoring Matrix



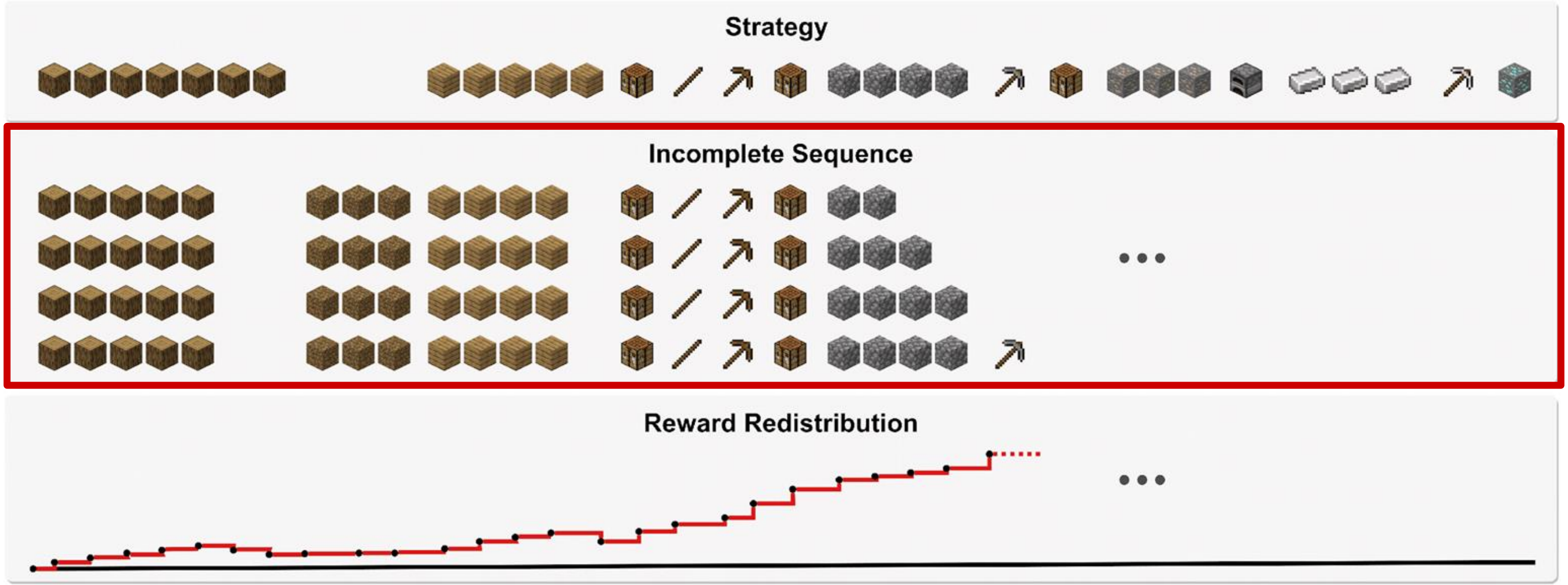
(IV) Position-specific Scoring Matrix



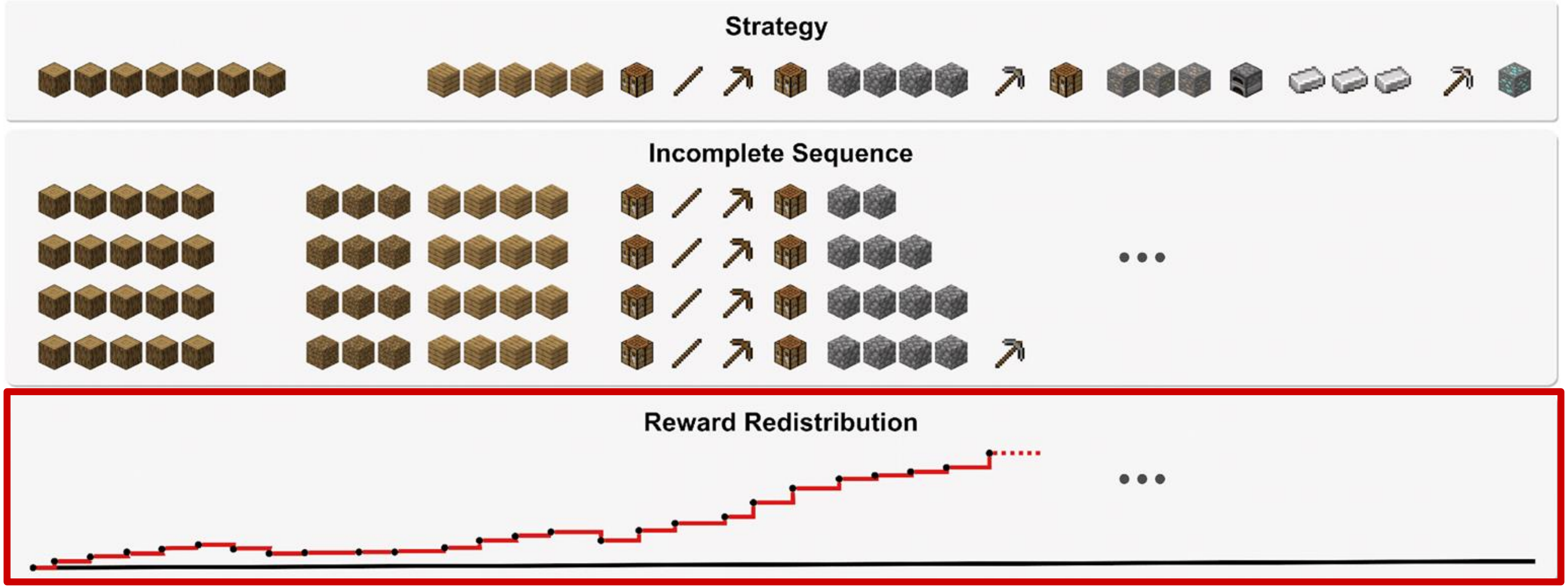
(IV) Reward Redistribution



(IV) Reward Redistribution

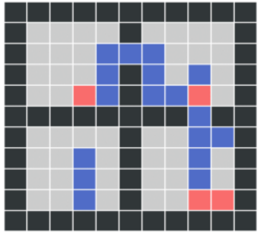


(IV) Reward Redistribution

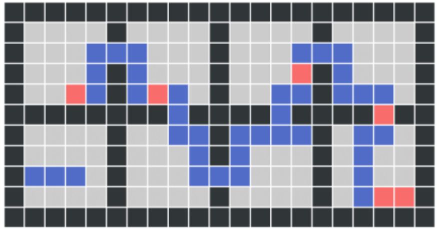


Experiments

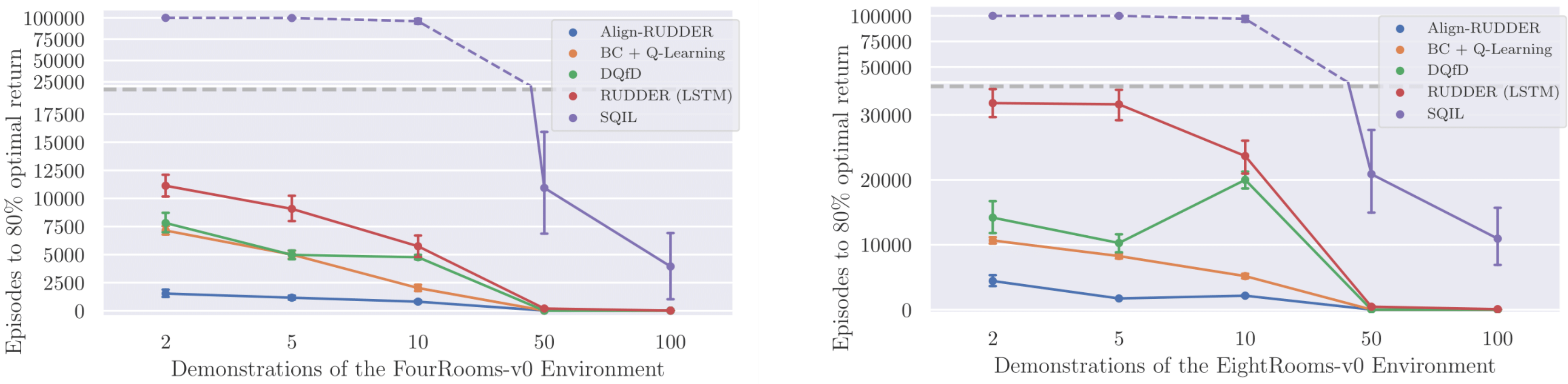
Experiments: Gridworld



Example of a reward redistribution in a grid world with four rooms



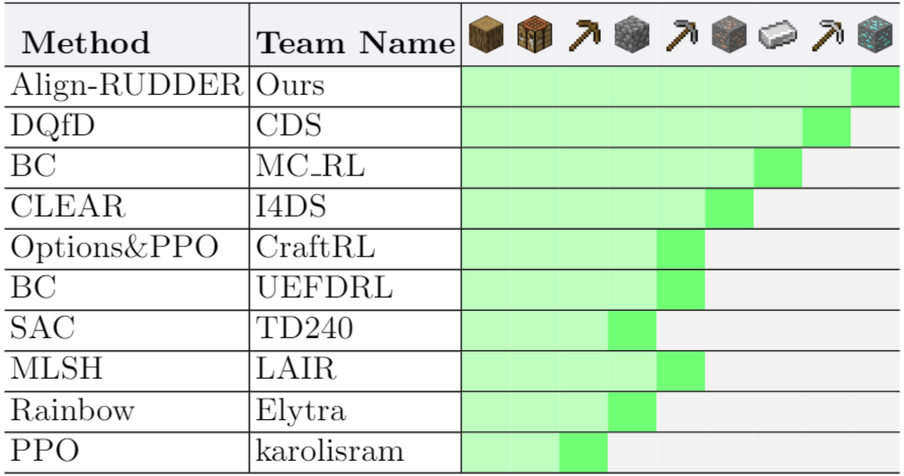
Example of a reward redistribution in a grid world with eight rooms



Comparison of Align-RUDDER to other methods with respect to the number of episodes required for learning on different numbers of demonstrations

Experiments: Minecraft

- First pure learning method to obtain a diamond in the MineRL environment
- Only 10 demonstrations were necessary to identify key events
- Hierarchical RL of sub-agents identified using reward redistribution



Method	Team Name									
Align-RUDDER	Ours	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
DQfD	CDS	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	
BC	MC_RL	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
CLEAR	I4DS	🟩	🟩	🟩	🟩	🟩	🟩			
Options&PPO	CraftRL	🟩	🟩	🟩	🟩	🟩				
BC	UEFDRL	🟩	🟩	🟩	🟩	🟩				
SAC	TD240	🟩	🟩	🟩	🟩					
MLSH	LAIR	🟩	🟩	🟩	🟩	🟩				
Rainbow	Elytra	🟩	🟩	🟩	🟩					
PPO	karolisram	🟩	🟩	🟩						

Contributions

SCAN ME

🐦: https://twitter.com/wehungpatil
🐦: https://twitter.com/mrkhof
📄: https://arxiv.org/abs/2009.14108
📜: https://ml-jku.github.io/align-rudder
💾: https://github.com/ml-jku/align-rudder
📹: https://tinyurl.com/2p8cdrfk

JKU JOHANNES KEPLER UNIVERSITY LINZ

enliteAI

dynatrace

Microsoft Research

IARAI institute of advanced research in artificial intelligence

- We suggest a reinforcement algorithm that works well for **sparse and delayed rewards**, where standard **exploration fails**
- We adopt **multiple sequence alignment** from bioinformatics to construct a reward redistribution technique that works with **few demonstrations**
- We propose a method that uses alignment techniques and reward redistribution for **identifying sub-goals and sub-tasks** which in turn allow for hierarchical reinforcement learning

[1] R. S. Sutton, ‘Temporal Credit assignment in Reinforcement Learning’, 1984
[2] Arjona-Medina et. al, ‘RUDDER: Return decomposition for delayed reward’, 2019
[3] Guss et. al, ‘The MineRL 2019 Competition on Sample Efficient Reinforcement Learning using Human Priors’, 2019