https://arxiv.org/abs/2005.06247

Explainable Reinforcement Learning: A Survey

Doyun Kim

XRL = XAI + DRL

XRL taxonomy

Table 1. Selected XRL methods and their categorization according to the taxonomy described in section 2.

Time \ Scope	Global	Local
Intrinsic	• **PIRL** (Verma et al. [63]) • Fuzzy RL policies (Hein et al. [22])	• **Hierarchical Policies** (Shu et al. [54])
Post-hoc	• Genetic Programming (Hein et al. [23]) • Reward Decomposition (Juozapaitis et al. [27]) • Expected Consequences (van der Waa et al. [64]) • Soft Decision Trees (Coppens et al. [8]) • Deep Q-Networks (Zahavy et al. [66]) • Autonomous Policy Explanation (Hayes and Shah [21]) • Policy Distillation (Rusu et al. [51]) • **Linear Model U-Trees** (Liu et al. [38])	• Interestingness Elements (Sequeira and Gervasio [53]) • Autonomous Self-Explanation (Fukuchi et al. [17]) • **Structural Causal Model** (Madumal et al. [41]) • Complementary RL (Lee [33]) • Expected Consequences (van der Waa et al. [64]) • Soft Decision Trees (Coppens et al. [8]) • **Linear Model U-Trees** (Liu et al. [38])

Notes. Methods in bold are presented in detail in this work.

[63] Verma, A., Murali, V., Singh, R., Kohli, P., Chaudhuri, S.: Programmatically interpretable reinforcement learning. PMLR 80:5045-5054 (2018)

[54] Shu, T., Xiong, C., Socher, R.: Hierarchical and interpretable skill acquisition in multi-task reinforcement learning (2017)

[38] Liu, G., Schulte, O., Zhu, W., Li, Q.: Toward interpretable deep reinforcement learning with linear model u-trees. In: Machine Learning and Knowledge Discovery in Databases, pp. 414–429. Springer International Publishing (2019)

[41] Madumal, P., Miller, T., Sonenberg, L., Vetere, F.: Explainable reinforcement learning through a causal lens (2019)

Multi-level hierarchical policy

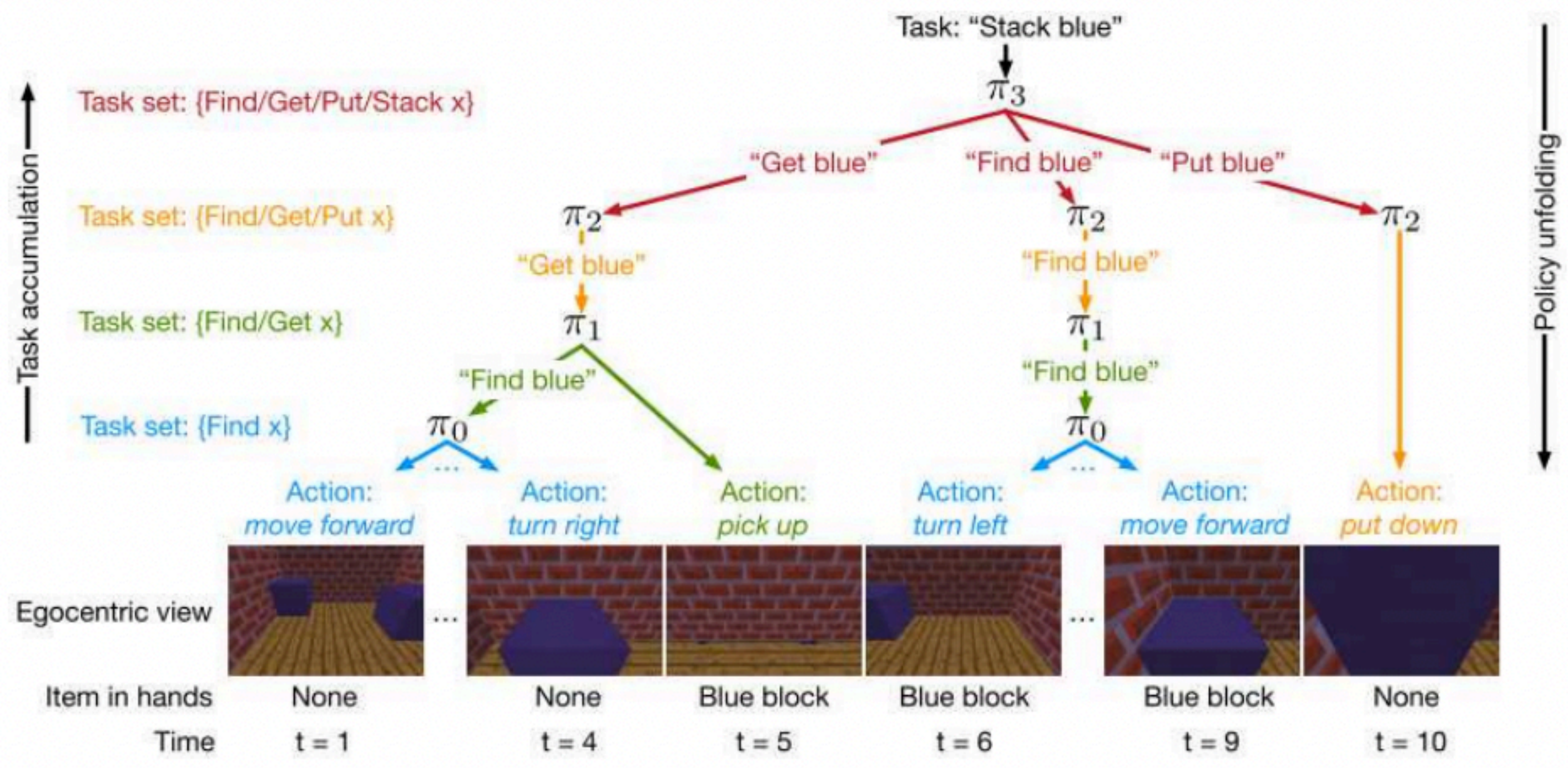


Fig. 5. Example for the mutli-level hierarchical policy for the task to stack two blue boxes on top of each other. The top-level policy (π_3, in red) encompasses the high-level plan 'get blue'→'find blue'→'put blue'. Each step (i.e., arrow) either initiates another policy (marked by a different color) or directly executes an action. Adopted from [54].

Linear Model U-Tree Training

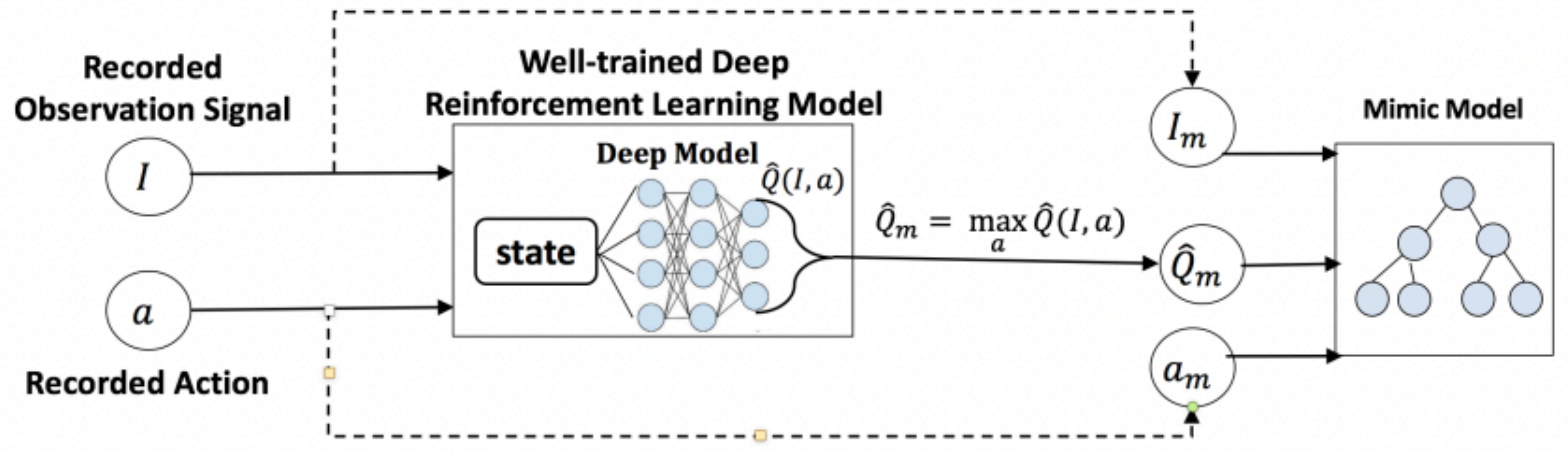


Fig. 1: Experience Training Setting

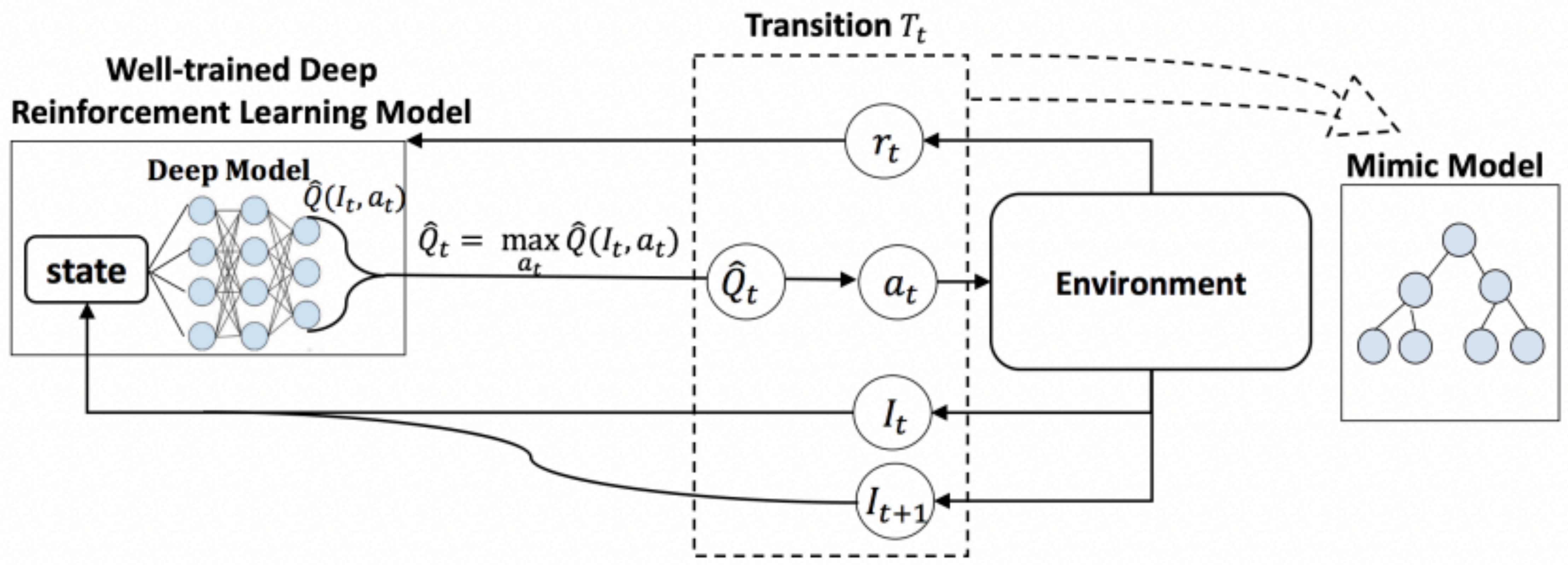


Fig. 2: Active Play Setting.

Linear Model U-Tree Structure

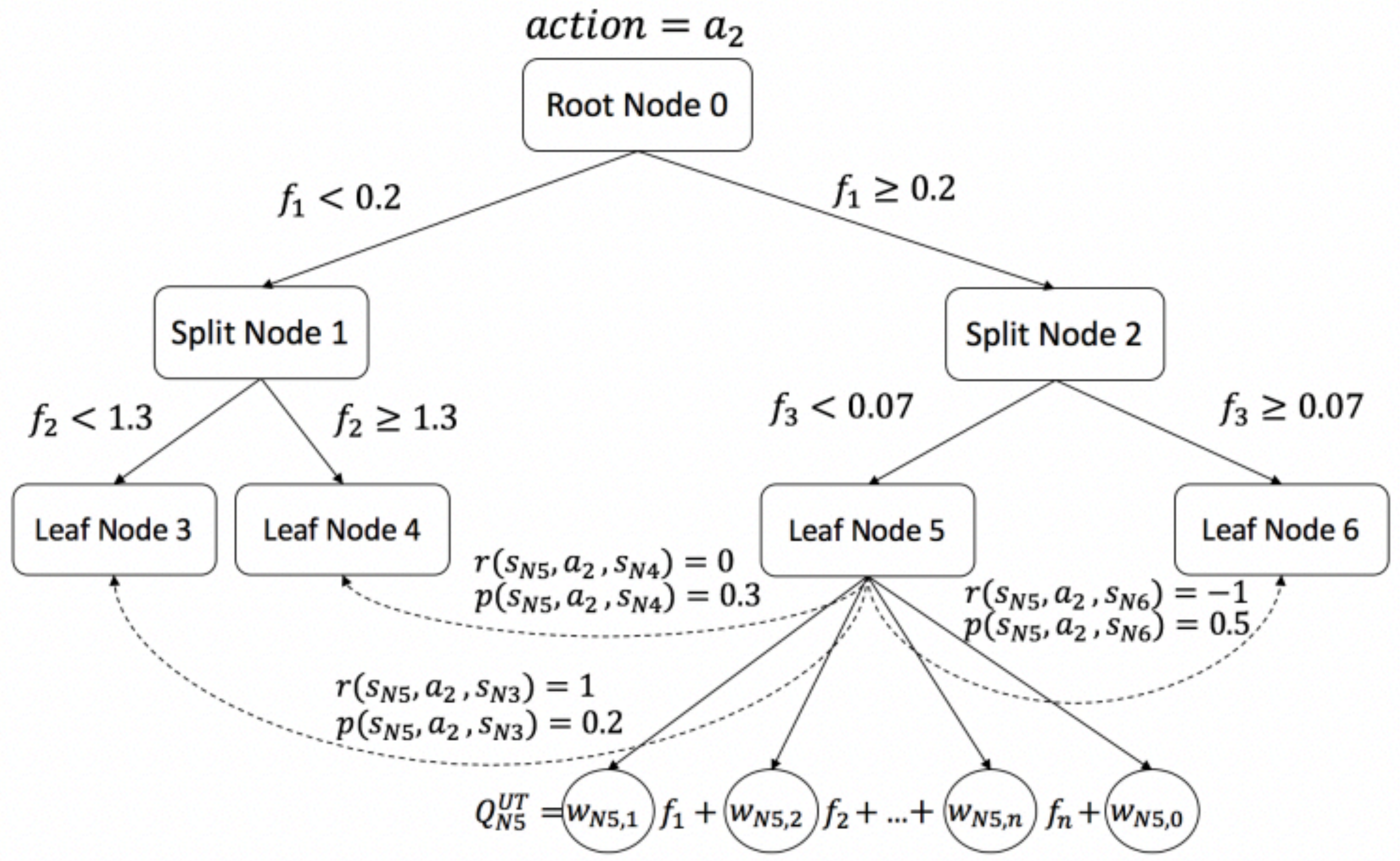


Fig. 3: An example of Linear Model U-Tree (LMUT).

Table 1: Partition Cell

Node Name	Partition Cell
Leaf Node 3	$f_1 < 0.2$, $f_2 < 1.3$
Leaf Node 4	$f_1 < 0.2$, $f_2 \geq 1.3$
Leaf Node 5	$f_1 \geq 0.2$, $f_2 < 0.07$
Leaf Node 6	$f_1 \geq 0.2$, $f_2 \geq 0.07$

Rule extractor

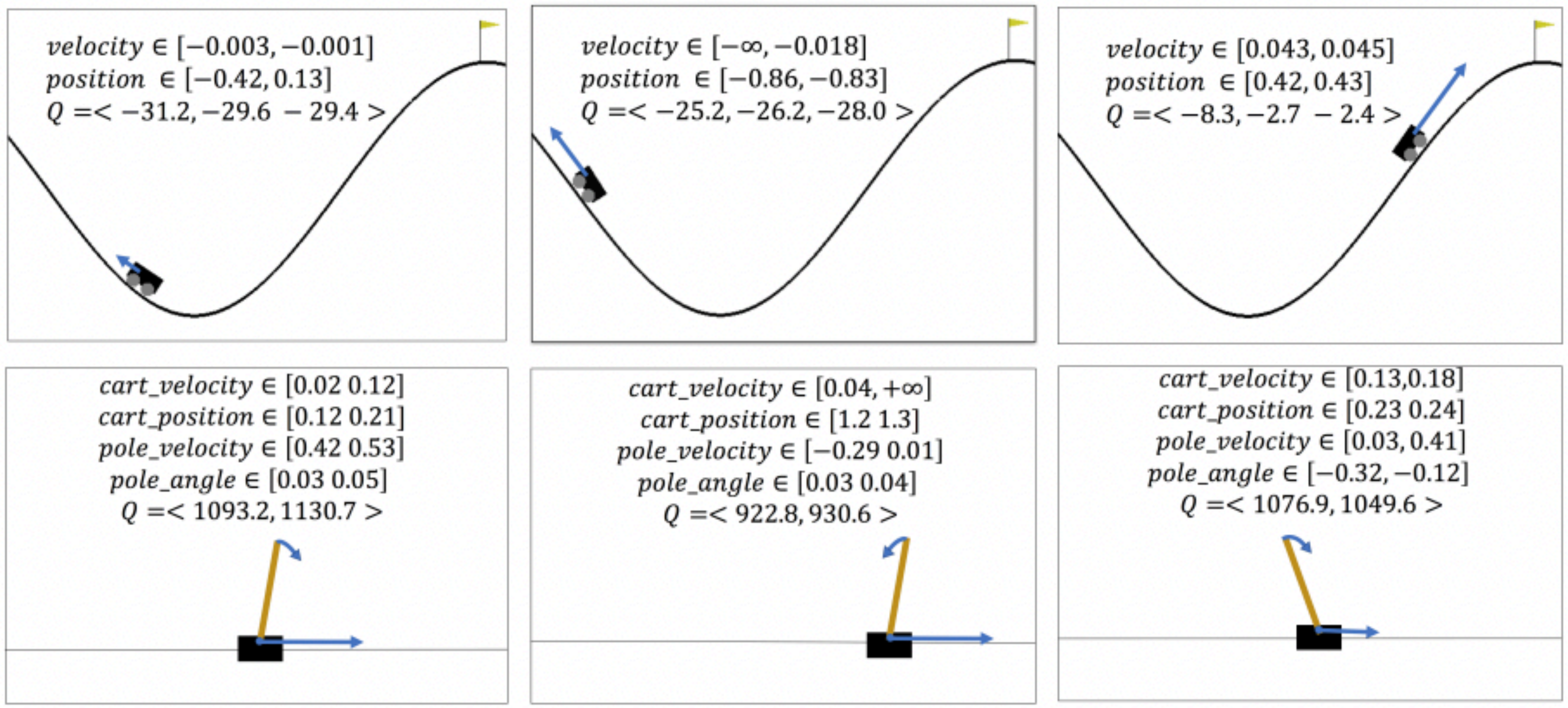
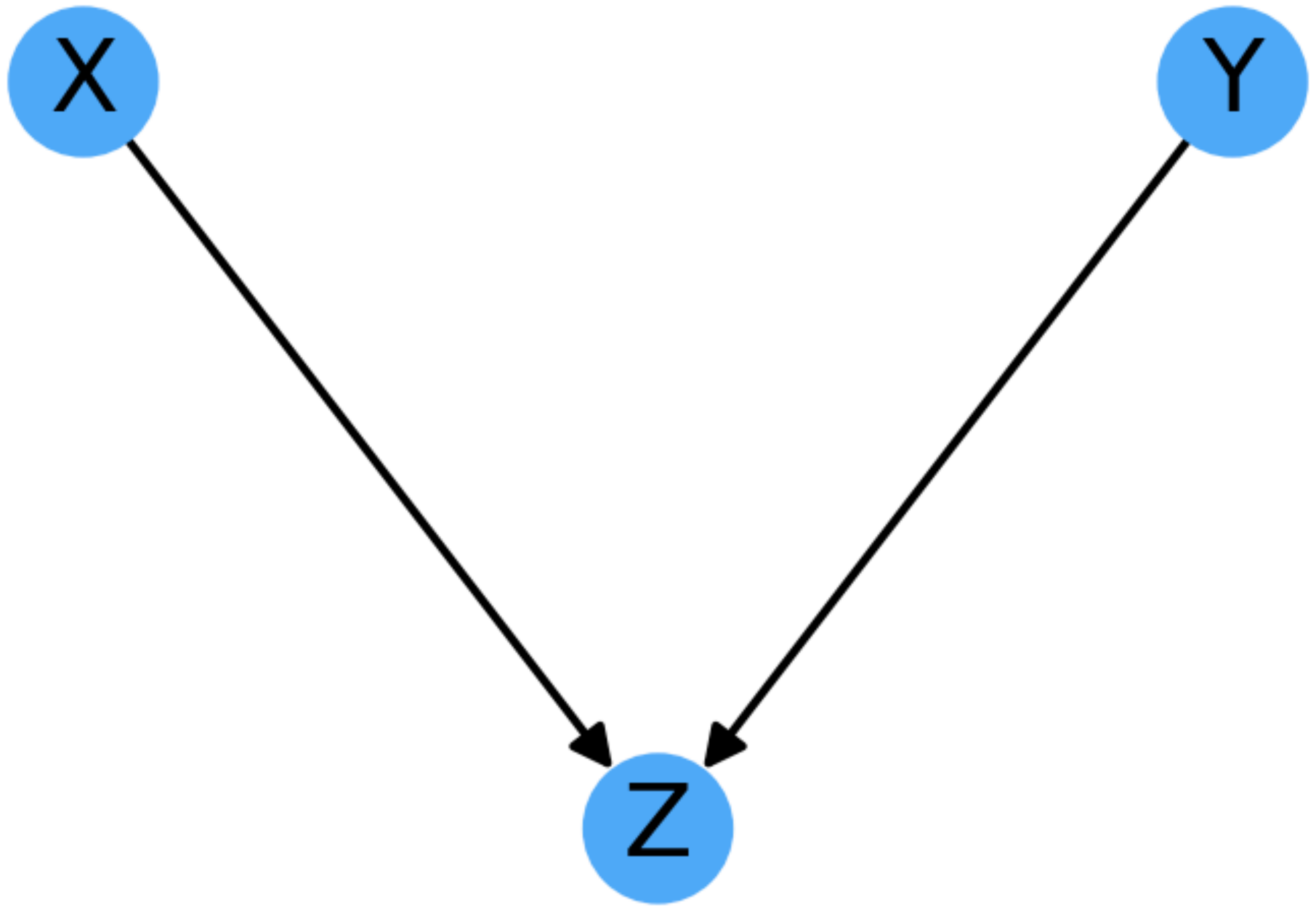


Fig. 7: Examples of Rule Extraction for Mountain Car and Cart Pole.

Structural Causal Model (1)



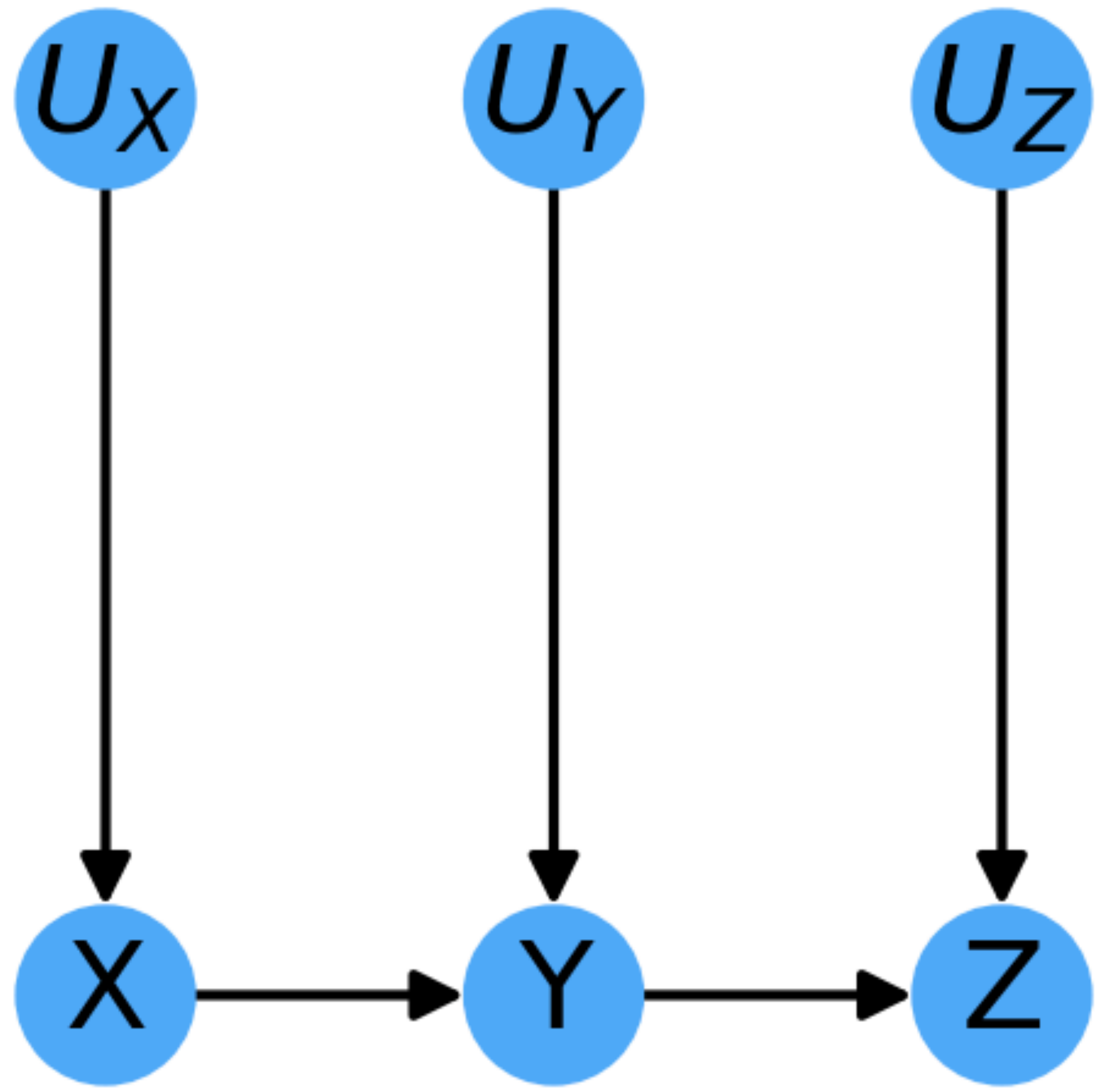
$$
\begin{aligned}
U &= \{X, Y\} \\
V &= \{Z\} \\
F &= \{f_Z : \ Z = 2X + 3Y\}
\end{aligned}
$$

U: Exogenous (External) Incoming edge를 가지고 있지 않음

V: Endogenous (Internal) Incoming edge를 가지고 있음

F: Structural Equation (UuV)에서 V로 가는 관계

Structural Causal Model (2)



$$
\begin{aligned}
U &= \{U_X, U_Y, U_Z\} \\
V &= \{X, Y, Z\} \\
F &= \{f_X, f_Y, f_Z\} \\
f_X &: \quad X = u_X \\
f_Y &: \quad Y = \frac{X}{3} + U_Y \\
f_Z &: \quad Z = \frac{Y}{16} + U_Z
\end{aligned}
$$

U: Exogenous (External) Incoming edge를 가지고 있지 않음

V: Endogenous (Internal) Incoming edge를 가지고 있음

F: Structural Equation (UuV)에서 V로 가는 관계

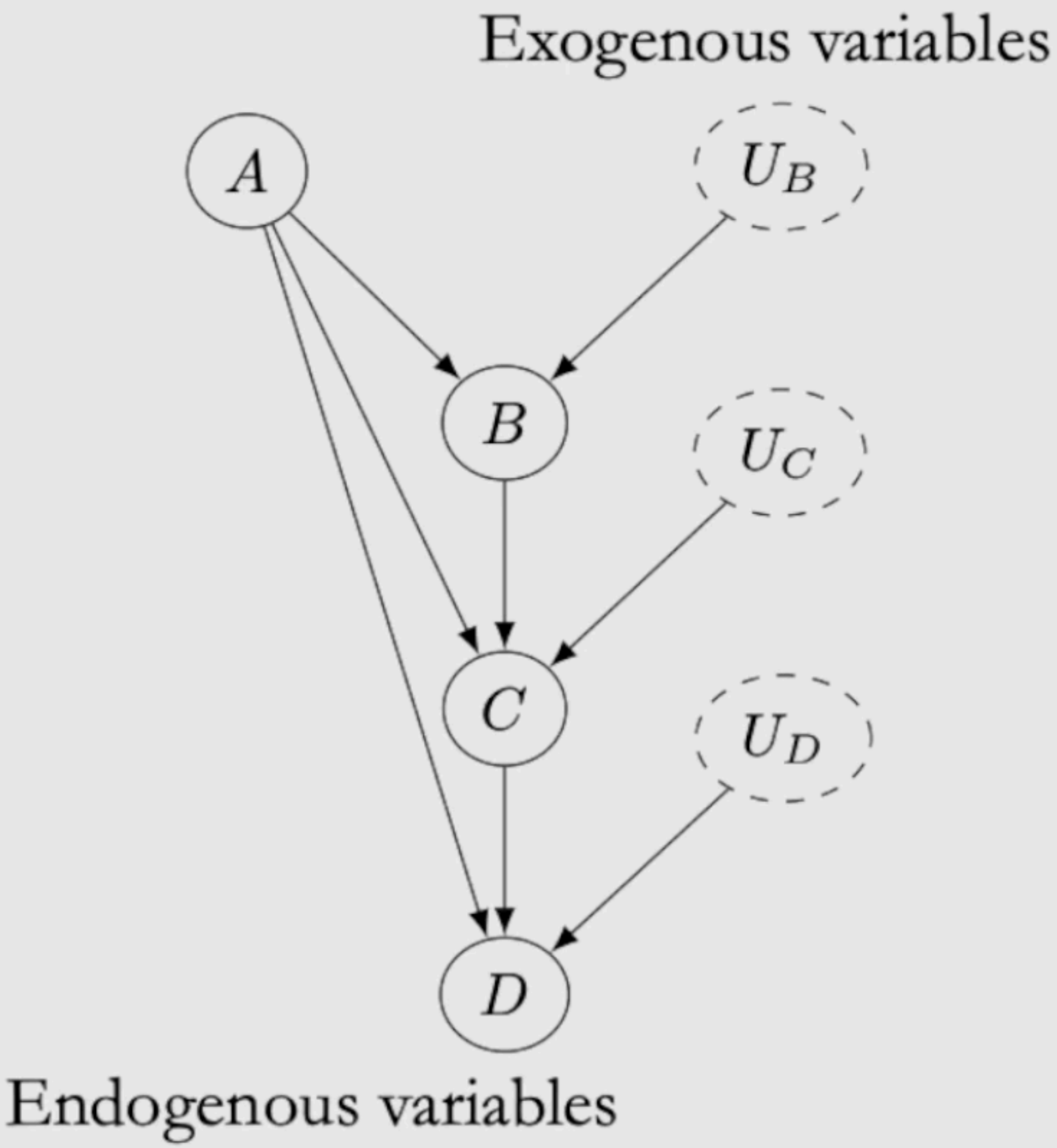
Structural causal models (SCMs)

$$M: \quad \begin{aligned} B &:= f_B(A, U_B) \\ C &:= f_C(A, B, U_C) \\ D &:= f_D(A, C, U_D) \end{aligned}$$

SCM Definition

A tuple of the following sets:

1. A set of endogenous variables
2. A set of exogenous variables
3. A set of functions, one to generate each endogenous variable as a function of the other variables



Action Influence model

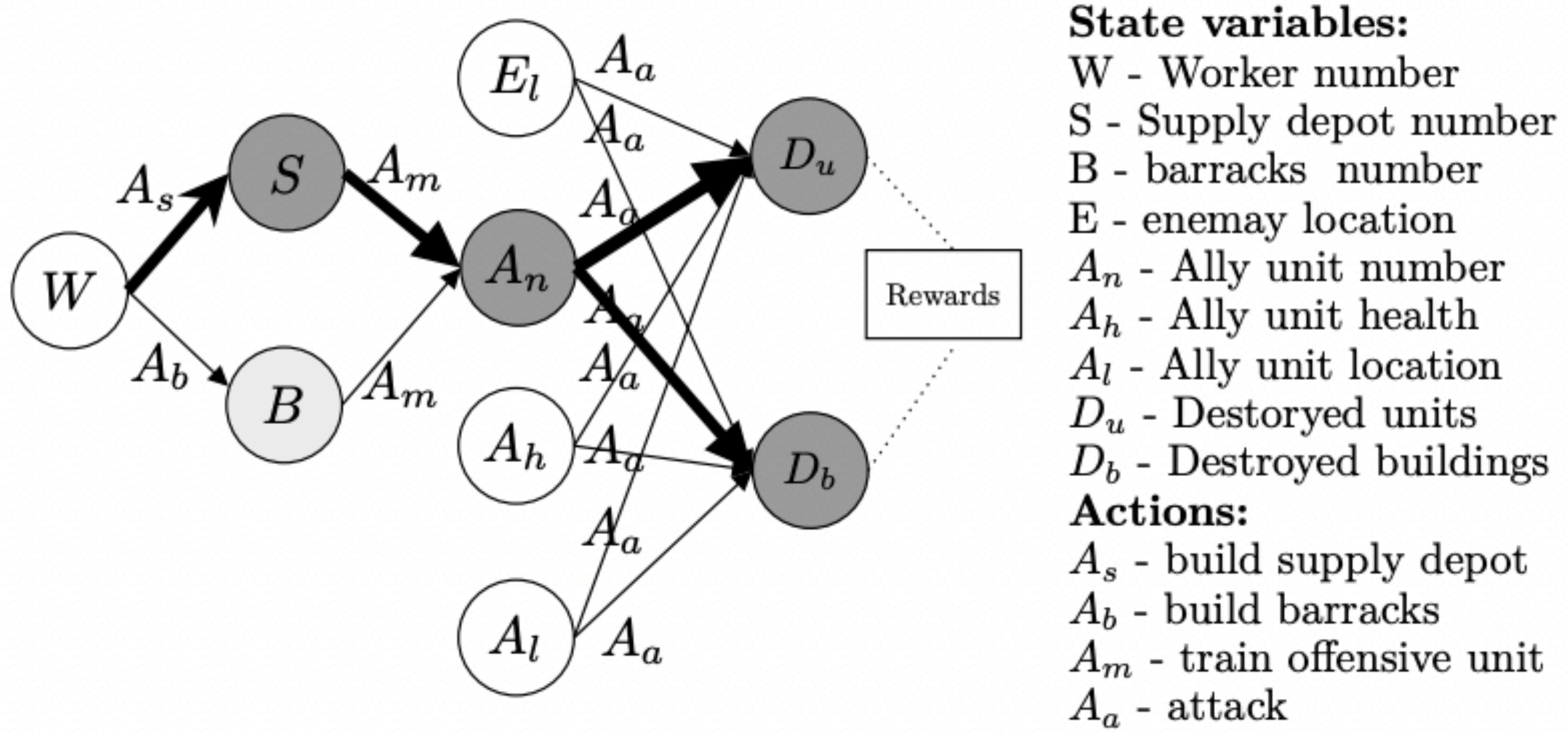


Figure 1: Action influence graph of a Starcraft II agent

Example 1. Consider the question, asking why a Starcraft II agent built supply depots, rather than choosing to build barracks:

Question	Why not *build_barracks* (A_b)?
Explanation	Because it is more desirable to do action build_supply_depot (A_s) to have more Supply Depots (S) as the goal is to have more Destroyed Units (D_u) and Destroyed buildings (D_b).

m = [W = 12, S = 1, B = 2, An = 22, Du = 10, Db = 7]

m’ = [W = 12, S = 3, B = 2, An = 22, Du = 10, Db = 7]