# Summarization: Explicitly Encouraging Low Fractional Dimensional Trajectories via Reinforcement Learning

Summarized by Xiaonan (Nice) Wang *
Summarization† Generated based on Paper from Gillen and Byl. (2021)
Topic: Legged Locomotion, Fractal Geometry, Reinforcement Learning

April 1, 2026


### Abstract

This document summarizes the core contributions and methodology of the paper "Explicitly Encouraging Low Fractional Dimensional Trajectories via Reinforcement Learning, Gillen and Byl [2021]", focusing on its' main ideas and the core blocks.


## 1 Overview: Core Questions and Answers

#### (1) What is the problem?

- In *Legged Locomotion*: Learning an intelligent gait that possesses both **low-dimensional stability** and **high adaptability** within a **stochastic real-world** environment.
  - via **Fractional Dimensionality Reduction**,
  - facilitating **Formal Verification**,
  - and being **robust** to *Noise & Disturbances* (**stability** or **robustness**).

#### (2) Why need to solve this problem?

- **Robustness and Stability**: Guarantee the system is safe in all the regions of state space during its life.
- Breaking the **curse of dimensionality**: Dealing with scenarios with *high (topological) dimensional state space*.
- Addressing the **Unexplainable** and **Unverifiable** issues of **Model-Free approaches**.

#### (3) How is it different from prev.?

- When Box Meshing: Storing mesh points in a **Hash Table**.
  - Time Complexity: **Reducing** from $o(n^2)$ to $o(n)$;
  - **Data Compression**: $cnt(keys) \ll n$, points belong to same box will share the same key.
- When Post Processing: "**Divided** by fractional dimension" **rather** than "*Subtracted* by fractional dimension".

*wangxiaonannice@gmail.com
†**Disclaimer:** This summarization is for research and learning purposes. It represents a personal interpretation and may contain inaccuracies. Feedback or corrections via email are highly appreciated.

- Reward Scaling;
- Avoiding negative rewards: As negative signals can impede convergence of some RL, division is sign-preserving, compressing the reward magnitude without altering its polarity.

- **Clipped** Fractional Dimension (in post processing):
  - Force to $\geq 1$: Preventing agents learning to *fall over immediately* to "game the system";
  - Force to $\leq halfoftopologicaldimension$: Providing a numerically smooth "ceiling" for training to prevent gradients from *spiraling out of control* (*exploding*).

**(4) Why is it better than prev.? (Advantages)**

- **Low Fractional Dimension** and **More Stable** Trajectories
- Hash Table for Data Compression when Box Meshing
- **Computational Efficient** (Hash Table Box Meshing + Model-Free RL + Fractal Penalty)
- Pursuing Low-Dimensional Trajectories for **Formal Verification**

**(5) What is the approach itself?**

- Approach Itself:
  - Box Meshing
    - Hash(Mesh Points);
    - Using Hash Table to stire keys.
  - Fractional Dimension Calculation
    - Building $Hsize(statespace) - > 1$ ( basically a **decreasing set**)
    - Calculating the Upper & Lower Mesh (Fractal) Dim via the slope of log-log curve (or variation)
  - Post Processing
    - Clipped Mesh Dim
    - Reward **Penalty**: Reward <- Reward / Clip-Dim
- Novelties:
  - **Geometric Regularization**: It introduces fractal dimension as a differentiable reward penalty to steer the agent towards low-dimensional manifolds.
  - **Computational Efficiency**: It proposes a hash-based online box-counting algorithm that enables real-time complexity estimation.
  - **Provable Simplicity**: By actively breaking the "curse of dimensionality," the method produces policies that are not only high-performing but also amenable to formal verification.

**(6) What are the applications of it?** Legged Lomotion on robotic environments (*HalfCheetah-v2*, *Hopper-v2*, and *Walker2d-v2*) with high dimensionality (11-17 DOF). (Simulation only, real-world robot experiments needed.)

## 2 The Structure

**Summarized Block-Diagram** The summarized block-diagram of *Meshing Box RL* see fig. 1[1].

[1] For efficiency, this diagram was initially hand-drawn on paper and then converted into its current digital version using Nano Banana Pro.

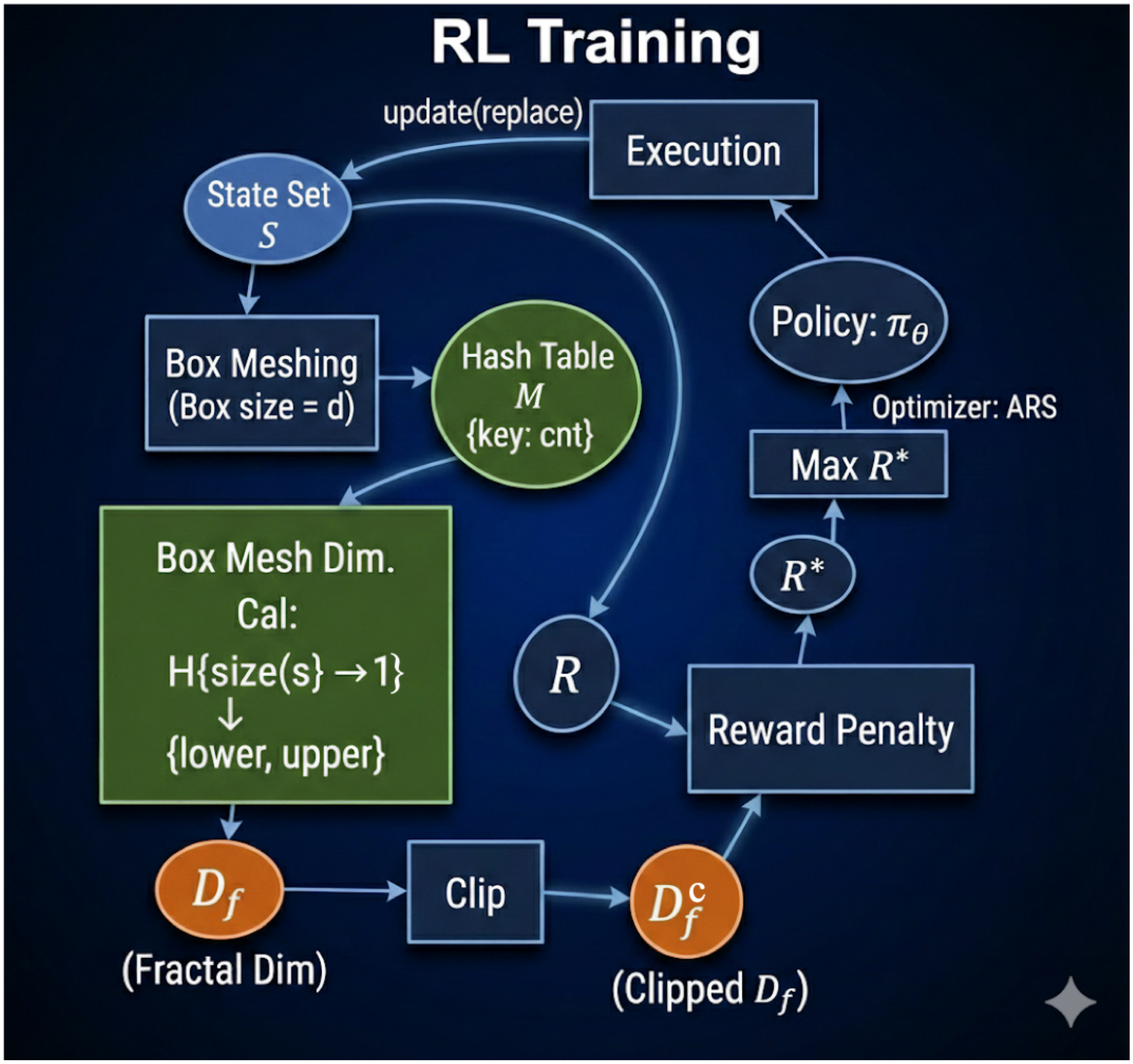


Figure 1: Summarized Block-Diagram of Meshing Box Reinforcement Learning

## 3 Open Questions

1. **Generalizing to Highly Stochastic Environments:** In highly stochastic environments, such as rocky terrains or slippery surfaces, complex maneuvers are often essential for survival. Does forcibly compressing the fractal dimension risk compromising the agent's performance in high-difficulty settings? Furthermore, is there an "optimal dimension" that strikes a balance between guaranteed verifiability and environmental adaptability?

2. **Beyond the Model-Free RL:** Beyond the field of Model-Free RL, does the concept of Fractal Regularization still hold significant value in legged locomotion based on recent popular approaches?

## References

S. Gillen and K. Byl. Explicitly encouraging low fractional dimensional trajectories via reinforcement learning. In *Conference on Robot Learning*, pages 2137–2147. PMLR, 2021.

## A Formula Derivation Draft[2]

The fig. 2 shows the "hand-crafted" formula derivation (draft) of Eq 1. when considering the special circumstance (i.e. considering topological dim.).

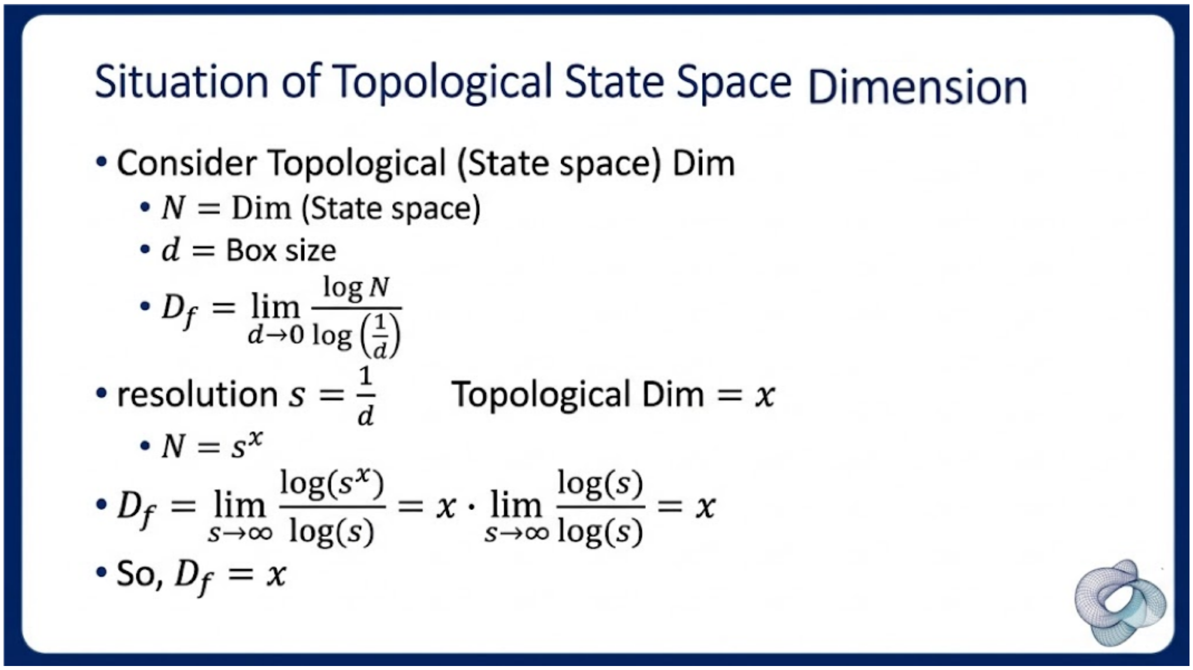


Figure 2: Formula Derivation: When considering about topological dim., Eq. 1 equals to topological dim.

## B Note: Physical DOF vs. Feature Dimension vs. Fractal Dimension

The fig. 3[2] shows the "hand-crafted *Mapping the Fea-Dimensionality to Multi-Dimensionality Coordinate Sys.*"

The table 1 illustrates the comparison of "*Physical DOF vs. State (Space) Dim. vs. Fractal Dim.*"

Table 1: Comparison of Dimensionality Concepts in Robot Locomotion

| **Concept** | **Explanation** | **Metaphor** | **Attribute** |
|---|---|---|---|
| **Physical DOF** | Degrees of Freedom | The number of **joints** (generally[3]) in the robot (hardware constraint). | Constant |
| **State Dimension** | Feature Dimension | **Total number of coordinate axes** describing the motion (typically $\geq 2\times$ DOF: e.g. for each joint, $\langle q, \dot{q} \rangle$ whereas $q$ is position and $\dot{q}$ is velocity). | Constant |
| **Fractal Dimension** | Fractal Dimension ($D_f$) | The **"thickness" or complexity** of the actual trajectory the robot follows. | **Variable**[4] |

[2]Converted by Nano Banana Pro.

[3]It's true that the physical DOF equals the number of joints or actuators in most robotic platforms (especially manipulators), but there are some *exceptions like: the DOF of UAV is 6 (x, y, z, roll, pitch, yaw) while with 4 actuators (for total/roll/pitch/yaw torque), the DOF of UGV is 3 (x, y, yaw) while with 2 actuators (left and right)*.

[4]Determined by RL policy, *theoretical maximum is State Space Dimension*.

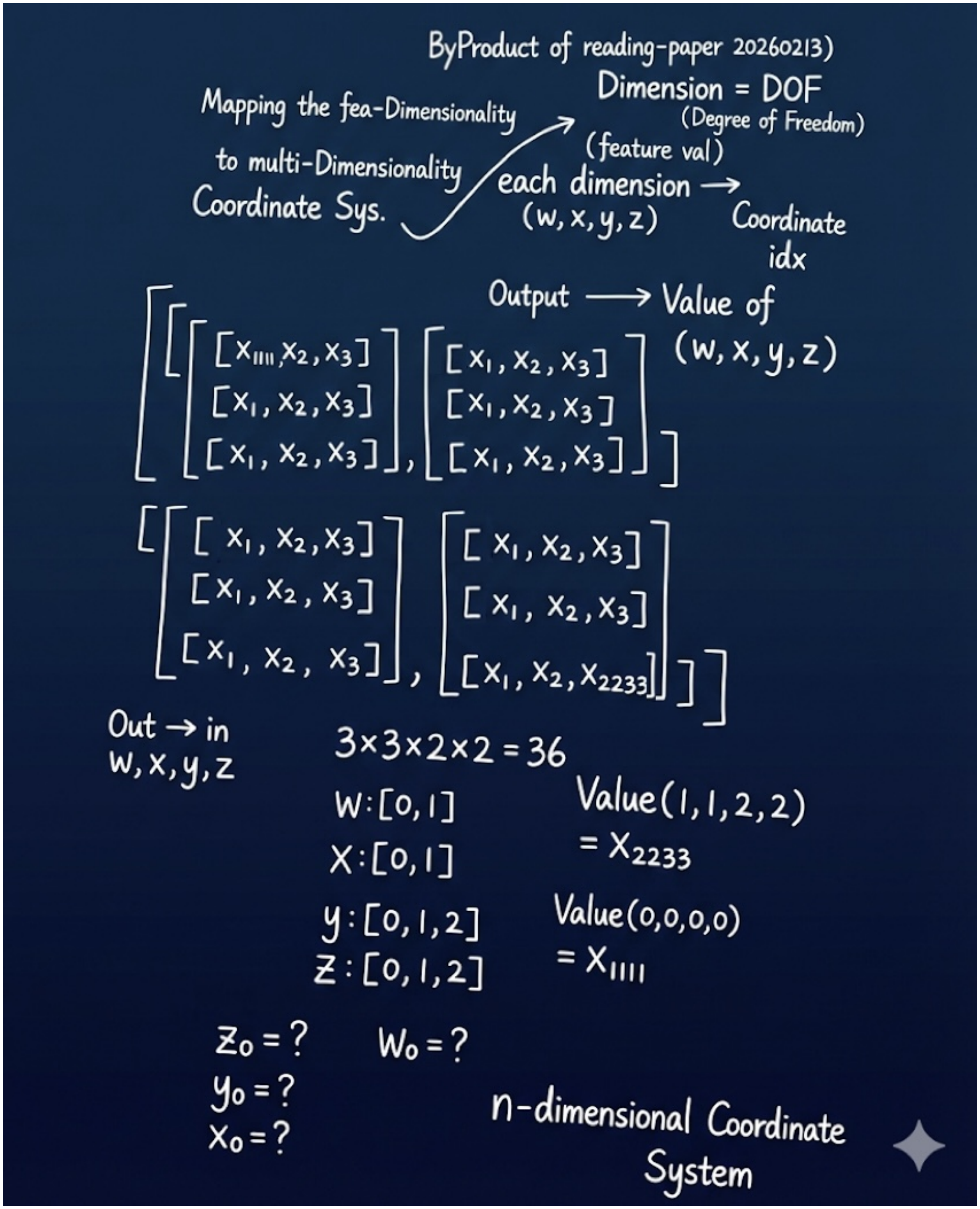


Figure 3: Note Draft: Mapping the Fea-Dimensionality to Multi-Dimensionality Coordinate Sys.