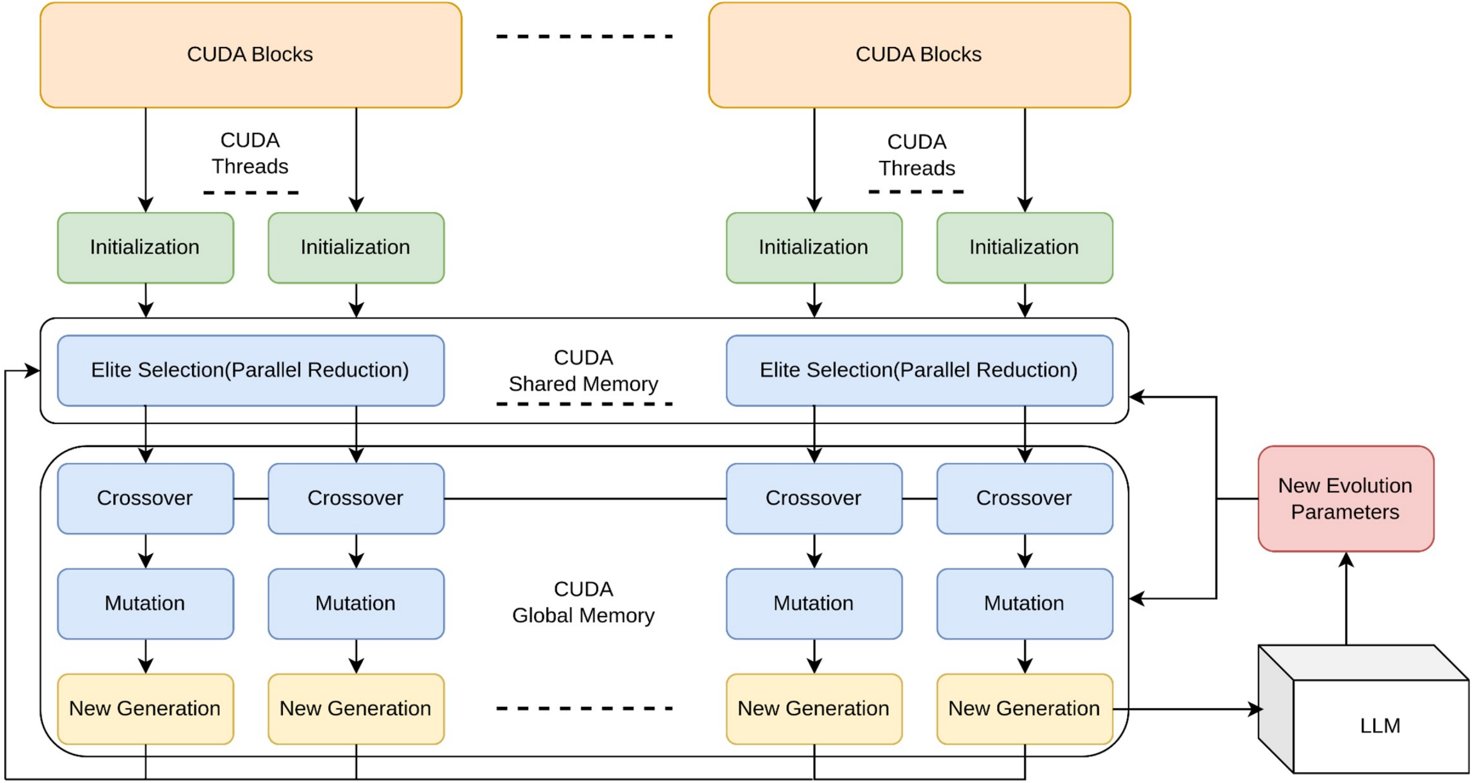
Evolving Soft Robots Using Implicit Neural Representation

MECS 4510 EVOLUTIONARY COMPUTATION AND DESIGN AUTOMATION

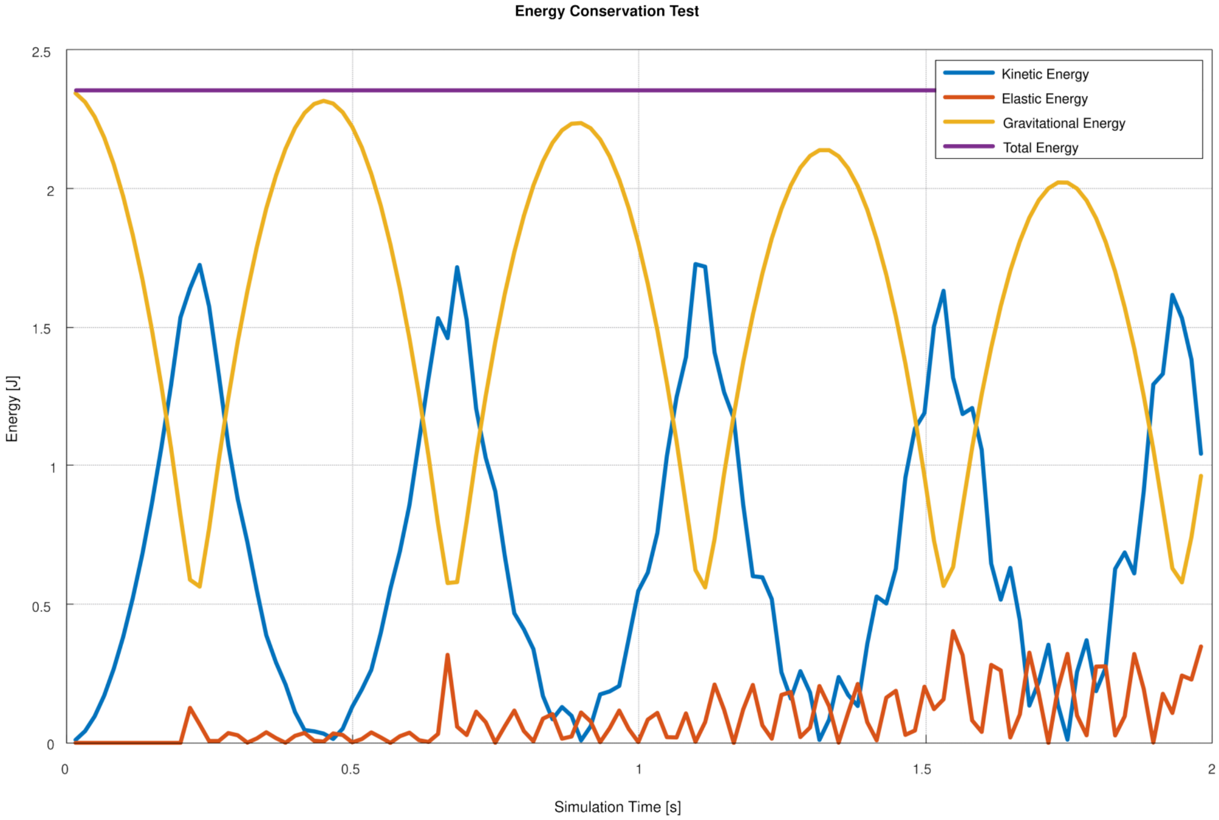
Name: Lechen Zhang
Uni: lz2954

Grace Hours Used: 80.5
Grace Hours Remaining: 15.5
Instructor: Hod Lipson

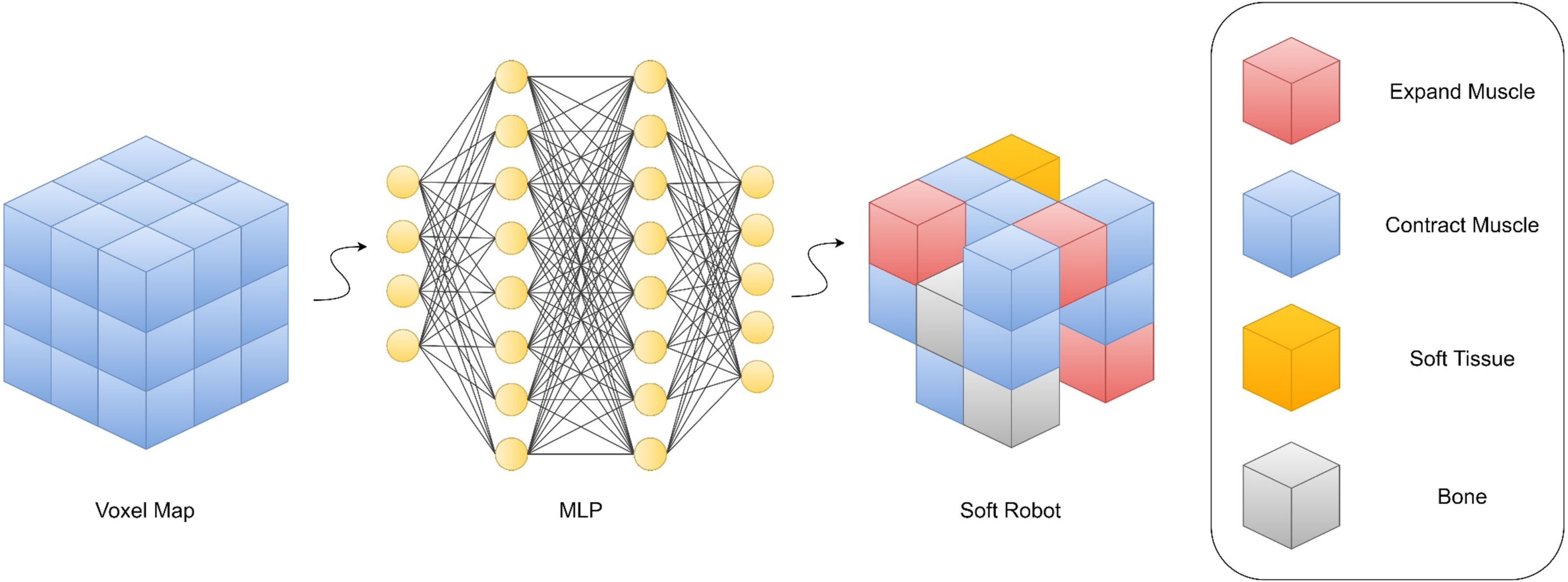
Overview


CUDA Blocks
CUDA Blocks
CUDA Threads
CUDA Threads
Initialization
Initialization
Initialization
Initialization
Elite Selection(Parallel Reduction)
CUDA Shared Memory
Elite Selection(Parallel Reduction)
Crossover
Crossover
Crossover
Crossover
New Evolution Parameters
Mutation
Mutation
CUDA Global Memory
Mutation
Mutation
New Generation
New Generation
New Generation
New Generation
LLM

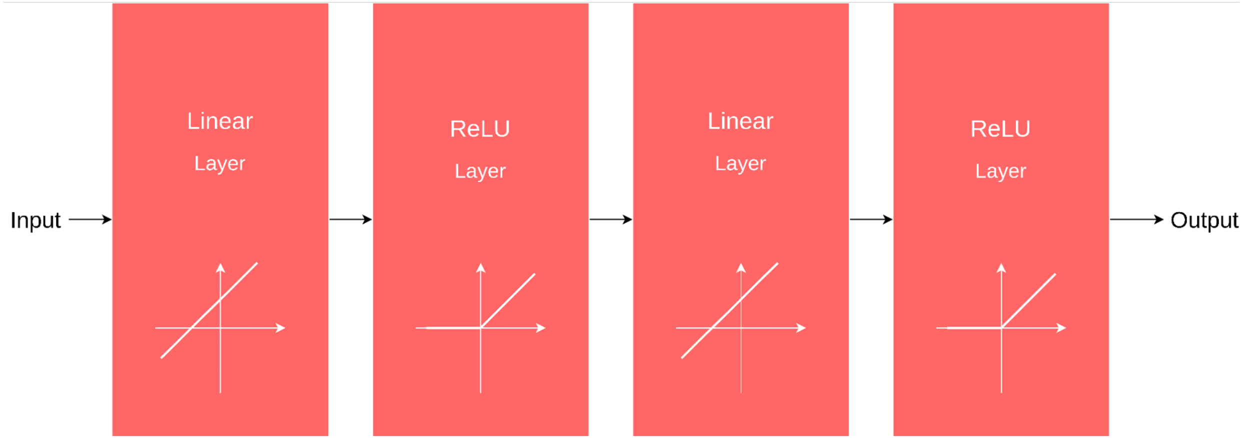
Simulation


Energy Conservation Test
Kinetic Energy
Elastic Energy
Gravitational Energy
Total Energy
Energy [J]
0
0.5
1
1.5
2
2.5
Simulation Time [s]
0
0.5
1
1.5
2

Voxel-Based Soft Robot



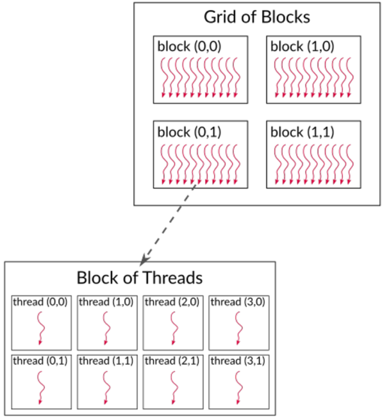
Multilayer Perceptron (MLP)



- Input: **(x, y, z, d)** The voxel position (x, y, z) and distance from the center (d)
- Linear Layer: Fully connected layer
- ReLU Layer: Activation layer
- Output : **(a0, a1, a2, a3, a4)** Biggest value decides the presence and material

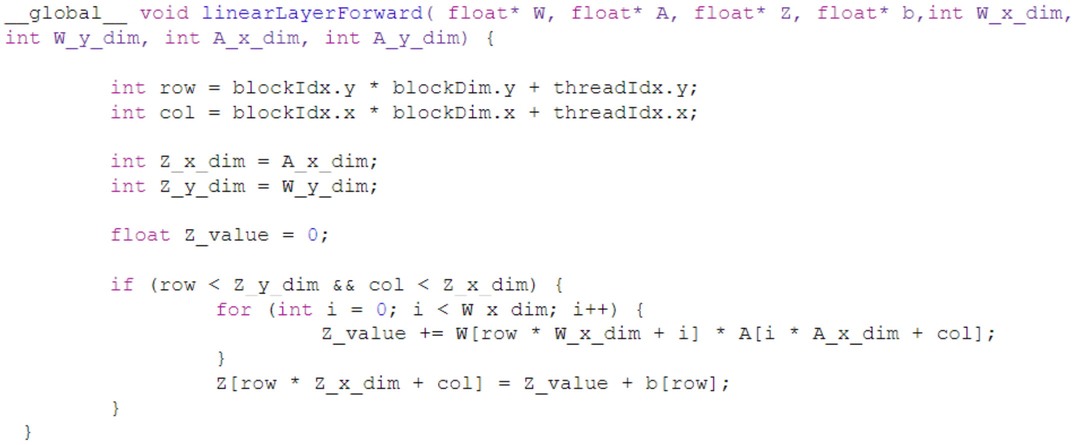
Accelerate Simulation via CUDA

```
void init(int num, bool on_host = true) {
             cudaError_t(*malloc)(void**, size_t);
             if (on_host) { malloc = &cudaMallocHost; }// if malloc =
cudaMallocHost: allocate on host
             else { malloc = &cudaMalloc; }// if malloc = cudaMalloc:
allocate on device
             gpuErrchk((*malloc)((void**)&k, num * sizeof(double)));
             gpuErrchk((*malloc)((void**)&rest, num * sizeof(double)));
             gpuErrchk((*malloc)((void**)&damping, num * sizeof(double)));
             gpuErrchk((*malloc)((void**)&left, num * sizeof(int)));
             gpuErrchk((*malloc)((void**)&right, num * sizeof(int)));
             gpuErrchk((*malloc)((void**)&isActuated, num * sizeof(bool)));
             gpuErrchk((*malloc)((void**)&actuator_coeff, num *
sizeof(Vec)));
             this->num = num;
       }
```



- Instead of allocating the entire Mass and Spring struct, an optimized memory management strategy is utilized that separately allocates the property arrays. This allows for fine-grained control over memory allocation and can potentially reducing memory waste.

Accelerate Forward Propagation via CUDA



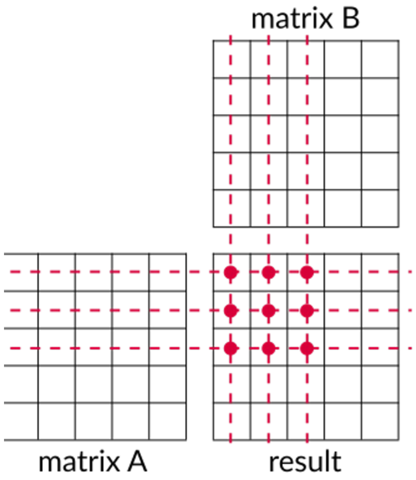
```
__global__ void linearLayerForward( float* W, float* A, float* Z, float* b,int W_x_dim,
int W_y_dim, int A_x_dim, int A_y_dim) {

        int row = blockIdx.y * blockDim.y + threadIdx.y;
        int col = blockIdx.x * blockDim.x + threadIdx.x;

        int Z_x_dim = A_x_dim;
        int Z_y_dim = W_y_dim;

        float Z_value = 0;

        if (row < Z_y_dim && col < Z_x_dim) {
                for (int i = 0; i < W_x_dim; i++) {
                        Z_value += W[row * W_x_dim + i] * A[i * A_x_dim + col];
                }
                Z[row * Z_x_dim + col] = Z_value + b[row];
        }
  }
```

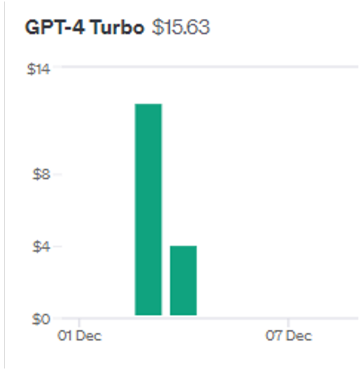


- In order to avoid race conditions, a parallel strategy that every single thread will compute a single pink dot. Every pink dot is a dot product of a row from matrix A and a column from matrix B.

Evolutionary Algorithms with LLMs

GPT-4 Turbo

- Expensive but support 128K tokens text window
- Monitor the evolution process and dynamically update the evolution parameters for better diversity and results

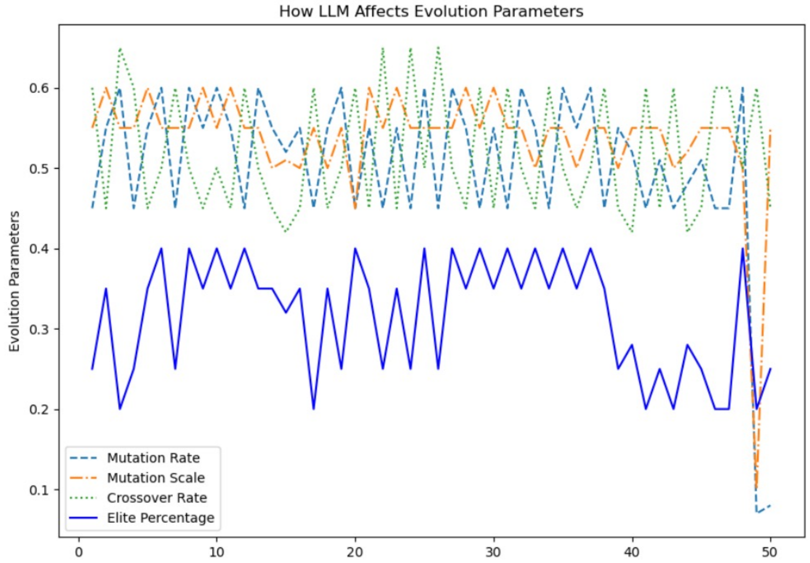


A 30 minus evolution with 50 generations costs 15.63$

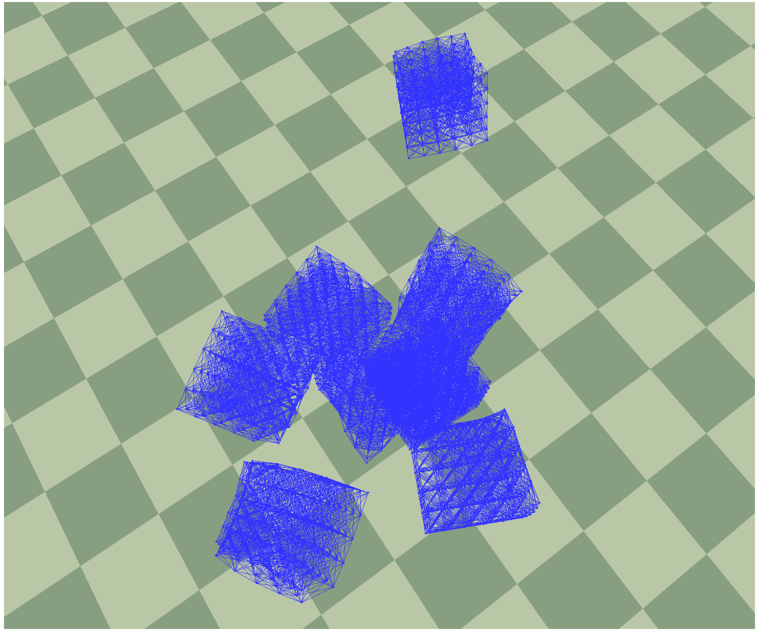
Diversity

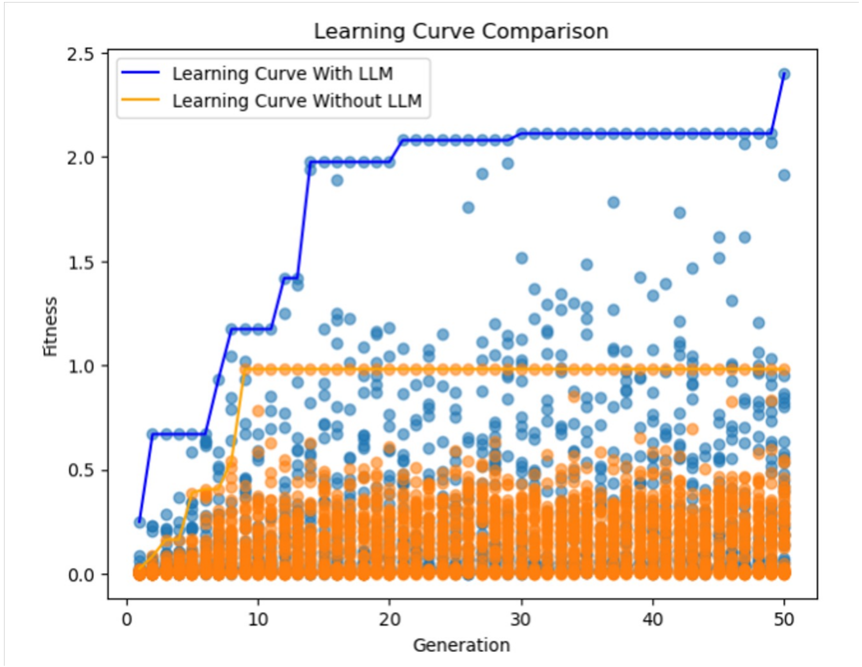
- Modeled by the average Hamming Distance of the voxel maps in one generation

- With the LLM supervision, diversity is well maintained throughout the evolution process



Results





- Generations: 50
- dT = 0.00001
- Population Size: 30
- Fastest Speed: 2.39879 m/s
- The Number of Springs Evaluated per Second: 2.8 * 10^9 (Approximately)

Future Works

Better Implicit Encoding and Neural Evolution

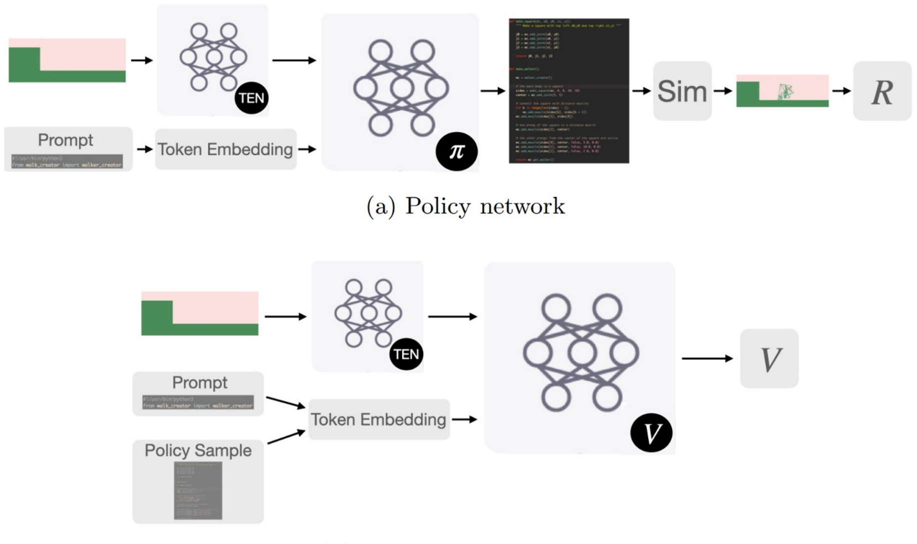
- HyperNEAT and ES-HyperNEAT in CUDA

Wiser Crossover Strategy

- Divide the population into species to achieve a more efficient crossover within the same species

Tightly Coupled LLM Evolutionary Algorithm

- Give the right to mutate and crossover certain parts of the code to LLMs.



(a) Policy network

(b) Value-function network

Reference List

1. J. Austin, R. Corrales-Fatou, S. Wyetzner and H. Lipson, "Titan: A Parallel Asynchronous Library for Multi-Agent and Soft-Body Robotics using NVIDIA CUDA," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 7754-7760, doi: 10.1109/ICRA40945.2020.9196808.

2. B. Xia, J. Fu, H. Zhu, Z. Song, Y. Jiang and H. Lipson, "A Legged Soft Robot Platform for Dynamic Locomotion," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 11812-11819, doi: 10.1109/ICRA48506.2021.9561018.

3. Zhang, Z. "Soft-Body Simulation With CUDA Based on Mass-Spring Model and Verlet Integration Scheme." Proceedings of the ASME 2020 International Mechanical Engineering Congress and Exposition. Volume 7A: Dynamics, Vibration, and Control. Virtual, Online. November 16–19, 2020. V07AT07A025. ASME. https://doi.org/10.1115/IMECE2020-23221

4. Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. 2014. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. SIGEVOlution 7, 1 (August 2014), 11–23. https://doi.org/10.1145/2661735.2661737

5. Lehman, J., Gordon, J., Jain, S., Ndousse, K., Yeh, C., Stanley, K.O. 2024. Evolution Through Large Models. In: Banzhaf, W., Machado, P., Zhang, M. (eds) Handbook of Evolutionary Machine Learning. Genetic and Evolutionary Computation. Springer, Singapore. https://doi.org/10.1007/978-981-99-3814-8_11

6. Tanaka, Fabio, and Claus Aranha. "Co-evolving morphology and control of soft robots using a single genome." 2022 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2022.

Thank you !