GLSO: Grammar-guided Latent Space Optimization for Sample-efficient Robot Design Automation

Jiaheng Hu, Julian Whitman, and Howie Choset

THE ROBOTICS INSTITUTE

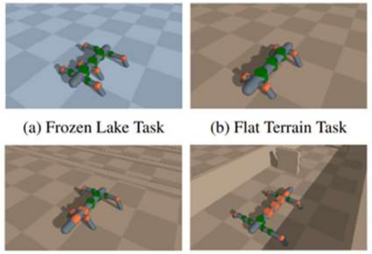
Biorobotics
Carnegie Mellon University

Introduction

Goal: automatically design robots specialized for a specific task, e.g., to locomote through a given environment.

Challenges:

- Sample-efficiency,
- Combinatorial explosion in number of possible designs,
- Diverse topology in design space,
- Cost of the evaluation of each design,
- Multi-modality in solutions.

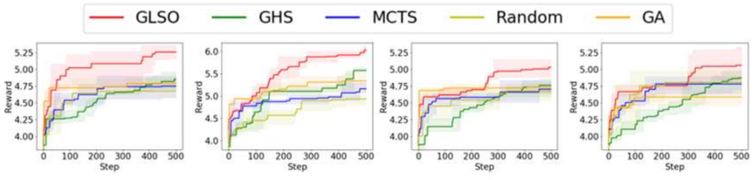


(a) Frozen Lake Task (b) Flat Terrain Task (c) Ridged Terrain Task (d) Wall Terrain Task

Robots optimized for four types of environments using our method

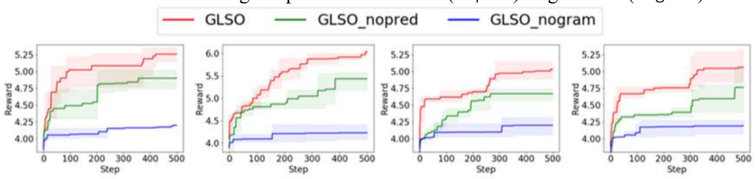
Experiments

Comparison with related approaches: Graph Heuristic Search (GHS), Monte-Carlo Tree Search (MTCS), Random Search, and Genetic Algorithm (GA) [1, 2, 3]. Ours (red line, averaged over three trials) finds designs with higher rewards in fewer samples.



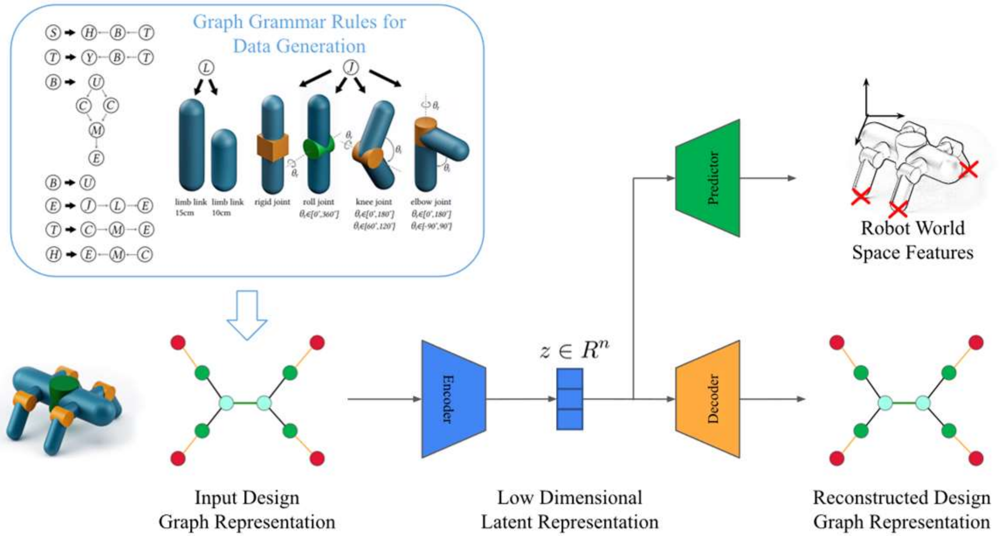
(a) Frozen Lake (b) Flat Terrain (c) Ridged Terrain (d) Wall Terrain

Ablation studies: removing the prediction network (nopred) or grammar (nogram)



(a) Frozen Lake (b) Flat Terrain (c) Ridged Terrain (d) Wall Terrain

GLSO Training Pipeline



Key idea:

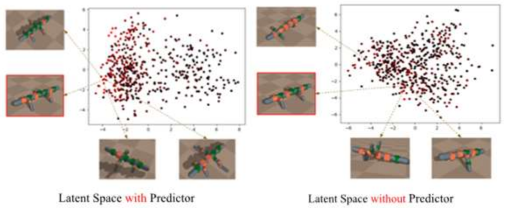
Transform the hard-to-search composition space into a meaningful continuous latent space and perform optimization in the latent space,

- **Graph VAE** for learning the transformation.
- **Graph grammar** for pre-eliminating invalid designs.
- **Property predictor network** encourages "distance" preservation in the latent space.
- **Bayesian Optimization** in the latent space to obtain designs.

Visualizing the Learned Latent Space

Interpolation between two points in the latent space

Comparison of learned latent spaces with / without the Property predictor network



Dirty Laundry

- Pre-defined graph grammar & world space feature selection.
- Extending to real-world hardware is non-trivial, e.g., difficult to verify real hardware optimality, sim-to-real gap.
- Sacrifices global optimality guarantee for sample-efficiency.

References

[1] A. Zhao, J. Xu, M. Konakovic-Lukovic, J. Hughes, A. Spielberg, D. Rus, and W. Matusik. " Robogrammar: graph grammar for terrain-optimized robot design, " ACM Transactions on Graphics (TOG), 2020.

[2] T. Wang, Y. Zhou, S. Fidler, and J. Ba. " Neural graph evolution: Towards efficient automatic robot design," Int. Conf. on Learning Representations (ICLR), 2018.

[3] G. S. Hornby, H. Lipson and J. B. Pollack, "Evolution of generative design systems for modular physical robots," Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2001