Exploring Inductive Bias in Model-Based RL for Inverse Material Design in Transistor Design

I compare the performance of deep Q-learning algorithms augmented with material dynamics models including graph-neural networks inspired by the SchNet architecture [11] that contain quantum chemical inductive biases under a distribution shift. The domain I focus on is the occupation and transport characteristics of silicon channels in transistors, which is related to my research in electrical engineering.

The reinforcement learning goal for this project is to identify the performance of a model-based DQN architecture under distribution shift. The motivation for this task is that training the model on the small environment is relatively easy. Sampling the small environment is theoretically cheap and feasible. If the model can learn a general representation from its experience with designing smaller structures, then when moved to the large environment it should require lower sample count to reach similar performance to a model that wasn't trained on the smaller environment. The main motivation here is to use a model-based approach. Literature shows that a sufficiently expressive model with the appropriate inductive biases (for example, an SchNet-based GNN) can learn how to derive the DOS from an input structure even for structures beyond the training set [12]. If a model pretrained on the smaller environment is moved to the larger dataset (smaller environment indicates structures are smaller, so simulation time is fast, while larger environment include complicated large structures), then the DQN can augment its training buffer with relatively accurate material property predictions from the pretrained environment model using rollouts from its environment sampling. This therefore reduces the number of environment samples necessary for the DQN network in the larger environment.

I generate data using a semiempirical simulation package made available through my lab (QuantumATK), which conducts quantum chemistry simulations. Database generation was conducted over a range of stresses varying the unit cell primitive vector from 0.9 to 1.035, with number of layers varying from 5 to 19. The database is split up in two in the following sections, with the one classified as the "small" environment data containing all structures with layers between 5 and 11, and the "large" environment data containing structures with layer count between 13 and 19.

I include the pretraining evaluation curves and deployment under a distribution shift in the larger environment for the SchNet model-based approach. While the small-environment pretraining doesn't show significantly lower sample complexity to reach high evaluation returns when compared to the model-free approach, we see a significantly higher evaluation return in the larger environment of the pretrained models, or equivalently a lower sample complexity to reach the same returns. Notably, the SchNet shows higher rewards than a dense dynamics model network and a model-free DQN in the larger environment. This demonstrates that pretraining a model-based system (notably one with appropriate inductive biases) for inverse material design could be an effective way to reduce sample complexity in novel chemical domains.

Introduction

Inverse material design is an application of quantum chemistry in which material property descriptions are converted into physical molecular or bulk structures [1]. Reinforcement learning algorithms for inverse material design have recently seen applicability and success over a wide range of domains including novel drug discovery [2, 5], organic molecule generation [3, 4], and inorganic crystalline solids [1]. The limiting factor in the application of reinforcement learning algorithms to many quantum chemistry problems, however, is the high computational cost of novel material simulation, which often requires costly semiempirical or *ab initio* simulation. Recent innovations in physics-inspired GNNs have shown success in predicting structural and electronic material properties for crystalline and amorphous materials [10], and have demonstrated success in generalizing to unseen chemical structures due to the model chemical inductive biases [12].

In this work, I aim to address the high sample complexity of exploration of novel chemical domains required to train inverse material design reinforcement learning algorithms by using a model-based approach. Specifically, I compare the performance of deep Q-learning algorithms augmented with material dynamics models including graph-neural networks inspired by the SchNet architecture [11] that contain quantum chemical inductive biases under a distribution shift. The domain I focus on is the occupation and transport characteristics of silicon channels in transistors, which is related to my research in electrical engineering. Design of transistor channels with tailored chemical properties is a holy grail in transistor research, and would enable low-energy, high-fidelity devices for next-generation transistor manufacturing.

Physical System and Simulation

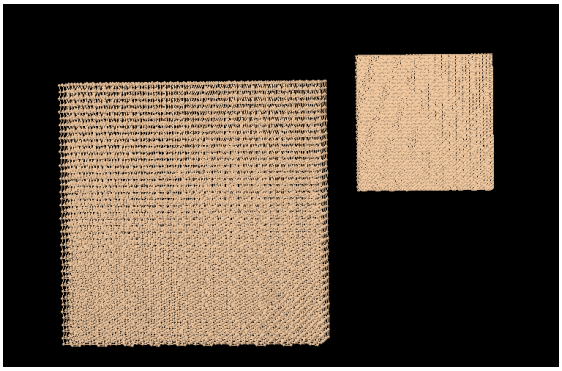

Fig 1. The chemical system of interest, a crystalline nanoslab of silicon (two are shown here)

System Overview

I focused on a model gate channel composed of crystalline elemental silicon, due to the ubiquity of the material in transistor design, and the relatively well-understood manufacturing properties of the material. Channel geometry is a nanoslab with diamond crystal structure, oriented in the [100] direction. The channel geometry can vary in two ways: the number of layers of crystalline silicon, and the stress on the crystal. Stress is applied isotropically to an approximately isotropic structure, leading to a simple rescaling of the lattice constant.

The material property that is predicted is the system density of states (DOS). The density of states determines the inversion charge characteristics and contributes ultimately to the conductivity of the transistor channel, which determines the on- and off-currents in the device. As a transistor functions as a current switch, the relative on- and off-currents are vital characteristics in a transistor. Thus, I use the density of states as the target characteristic of the material.

Models

Simulation was initially done in Python with custom code using Harrison Scaling with NRL overlap parameters [9], which was a large undertaking that consumed a large amount of the coding effort for this project. This system provided varied crystalline geometry and stresses, but made multiple unphysical approximations that ultimately led to a very noisy material property prediction. This led to a high-variance signal that correlated poorly with the physical material, which led to poor convergence characteristics for the model graph neural network.

I decided later to use a semiempirical simulation package made available through my lab (QuantumATK), which conducts quantum chemistry simulations. Specifically, I used the Bassani SiH Slater-Koster tight-binding simulation on the silicon nanoslab passivated with hydrogens on the dangling bonds, which led to density-of-states which were much closer to those found in literature. Database generation was conducted over a range of stresses varying the unit cell primitive vector from 0.9 to 1.035, with number of layers varying from 5 to 19. The database is split up in two in the following sections, with the one classified as the "small" environment data containing all structures with layers between 5 and 11, and the "large" environment data containing structures with layer count between 13 and 19. Runtime for the small database was approximately 4 hours. Runtime for the large data was approximately six days.

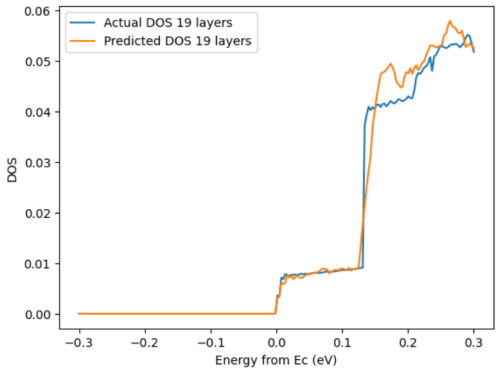

Fig 2. Example DOS from QuantumATK Database and GNN reproduction

Reinforcement Learning Architectures and Task

The reinforcement learning goal for this project is to identify the performance of a model-based DQN architecture under distribution shift. The motivation for this task is that training the model on the small environment is relatively easy. Database generation time is, as mentioned, around four hours, so sampling the small environment is theoretically cheap and feasible. If the model can learn a general representation from its experience with designing smaller structures, then when moved to the large environment it should require lower sample count to reach similar performance to a model that wasn't trained on the

smaller environment. The main motivation here is to use a model-based approach. Literature shows that a sufficiently expressive model with the appropriate inductive biases (for example, an SchNet-based GNN) can learn how to derive the DOS from an input structure even for structures beyond the training set [12]. If a model pretrained on the smaller environment is moved to the larger dataset (smaller indicates structures are smaller, so simulation time is fast, while larger are complicated large structures), then the DQN can augment its training buffer with relatively accurate material property predictions from the pretrained environment model using rollouts from its environment sampling. This therefore reduces the number of environment samples necessary for the DQN network in the larger environment.

Small Environment

Action: Discrete action space, size 4

- Increase/decrease lattice strain by 0.002 within range of 0.9 to 1.035
- Increase/decrease number of layers of the system by 1 with range of 5 layer to 11 layers

Observation: Concatenation of crystal strain, number of layers, and target density of states (evaluated at 200 points over a constant energy range)
Reward: If the target DOS is reached, reward is 1000. Otherwise, the reward is the negative Euclidean Norm of (target_DOS - current_DOS)^2
Actor Architecture: Deep Q-learning (no double-Q, since this led to destabilized training)
Model Architecture: Described in the next section, either a dense feed-forward layer, or an SchNet-based GNN

On reset, the initial structure state is randomized. Target states during training are also randomly chosen from the domain. The model replaces the "current_DOS" calculation with a neural network to greatly accelerate the reward signal calculation.

Large Environment

Action: Discrete action space, size 4

- Increase/decrease lattice strain by 0.002 within range of 0.9 to 1.035
- Increase/decrease number of layers of the system by 1 with range of 13 layer to 19 layers

Observation: Concatenation of crystal strain, number of layers, and target density of states (evaluated at 200 points over a constant energy range)
Reward: If the target DOS is reached, reward is 1000. Otherwise, the reward is the negative Euclidean Norm of (target_DOS - current_DOS)^2
Actor Architecture: Deep Q-learning (no double-Q, since this led to destabilized training)
Model Architecture: Described in the next section, either a dense feed-forward layer, or an SchNet-based GNN

On reset, the initial structure state is randomized. Target states during training are also randomly chosen from the domain. The model replaces the "current_DOS" calculation with a neural network to greatly accelerate the reward signal calculation.

Model Architecture

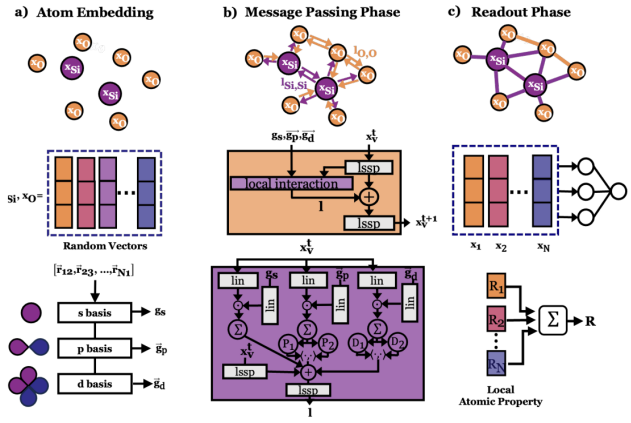

Fig 3. The graph neural network architecture [10]. This architecture includes the chemical inductive biases of locality, rotational invariance, rotational equivariance, molecular orbital spherical symmetries, and others.

The model graph neural network architecture is described in Fig 3. This model is inspired by the SchNet architecture [11], and modified for application for the DOS regime by work in the Salahuddin Lab. The graph structure is built over the atomic structure of the transistor. Connections are included over all neighboring atoms within a spherical cutoff of 5.0 Å. Node feature updates are continuous convolutions over the neighboring node features, in a message-passing neural network style [12]. More details on the architecture design are included in reference [10]. This model is predicted to handle the distribution shift from the small to the large environment well because of the model ansatz; it assumes locality of the chemical environment of each atom (which is physically motivated by Kohn et al [13]), which means it learns representations over chemical bonding environments rather than over structures. This means that when moving to a larger structure, it should theoretically be able to compose the local chemical environments of atoms to generate a good DOS prediction regardless of system size.

This argument has a few caveats, however. Notably, since our systems are so small (5-19 layers of atoms, which is tiny!), edge effects are very significant. This means that every atom effectively has a different chemical environment. However, this database itself took around a week of continuous compute to generate, so due to resource limitations I proceed with this. The following sections present the loss and error curves of the first iterations of the model-training step of the model-based DQN algorithm. The GNN is compared to a large dense feed-forward network for reference.

Model	Number of Parameters
Dense Small	17240
Dense Large	86760
SchNet Small	13053
SchNet Large	43585

Models Loss Curves over Distribution Shift

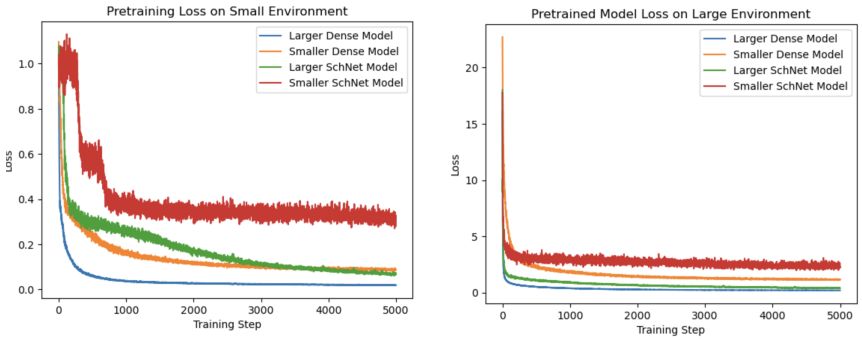

Fig 4. Loss Curves in the pretraining step in the small environment, and loss curves for the same model moved to the larger environment

Loss curves are included above to indicate the number of training steps required for a pretrained model to have low loss in the larger environment. I included a few sizes of neural network as an ablation study. Note that the loss scales shift as the target scales up by around a factor of ten, as DOS is larger for the larger systems. However, we see that convergence is achieved relatively quickly for the larger dense and SchNet models in the new environment, indicating that pretraining and deploying in a new environment may reduce sample complexity. For reference, a large SchNet model without having been pretrained required 2032 environment steps to each a loss below 1, as compared to 1137 for the large pretrained model.

Models Error over Layer Count over Distribution Shift

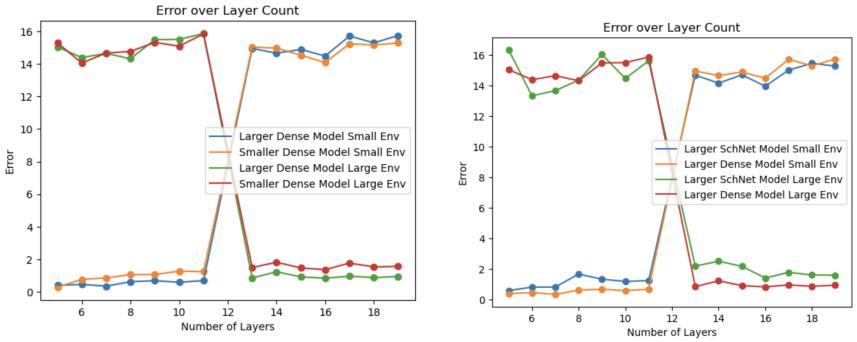

Fig 5. Error after training on small environment (curves labeled Small Env) or small environment then larger environment (curves labeled Large Env) for difference model configurations after one training iteration

The above curves demonstrate the error in the model under distribution shift. The curves labeled ‘small env’ indicate the errors over layer count for the dynamics model trained on the smalle environment (5-11 layers) before encountering larger structures (13-19 layers). Again, for ablation purposes, I compare a larger and smaller dense architecture to the SchNet architecture. We can see in the right figure that while the SchNet model doesn’t perform quite as well as the massive dense model in the training set, it marginally extrapolates better (has lower loss) in the unseen larger environment. Similarly, the smaller dense network does slightly better in the unseen environment. This implies that these models would do better in the unseen environment without any training.

The curves labeled ‘large env’ indicate the error after training the pretrained model on one iteration of samples from the larger environment. We can see that the smaller model has larger error after just a few samples along a random policy in the novel environment by around a factor of 10. Similarly, we see – contrary to our expectations – that the SchNet model has larger error than the simple dense network. There are a few reasons why this might be the case; one is that the domain is too simple for our hypothesis. While SchNet generalizes well across system sizes, the systems for the structures I made are quite similar, since, again, even small structures needed around a week to simulate with physical fidelity. Second is that the domain varies too quickly with layer count, since small-sized structures include a variety of chemical environments which are unseen. In fact, Kohn et al. noticed wavefunction interaction up to 30 Å for silicon, which is much larger than the 18 Å system thickness we’re considering. This indicates that the dense model might be sufficient for our applications of extrapolating to a new domain. The next sections show the evaluation returns using models based on the large dense network and the large SchNet model.

Results

Dense Dynamics Model

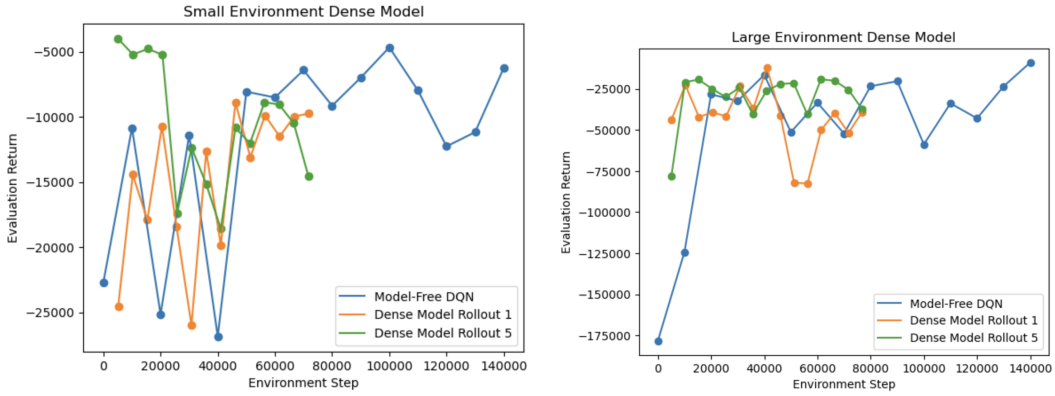

Fig 6. Pretraining and New Environment Deployment Step. Note that the model-free DQN has complete access to the system characteristics in the large environment, and therefore represents a theoretical maximum in asymptotic performance since the model-based approaches use reward signals derived from the noisy models to augment environment data

The above curves indicate the evaluation returns on the pretraining, small environment, and the subsequent evaluation returns moving the models to the larger environment using the large dense network from the previous section as the reward model. Two different rollout lengths are included for the model-based learning approaches as an ablation study. Note that the rollout length of five seems to have very high performance initially, but has unstable training. However, both the model-based approaches seem to perform well and reach the model-free asymptote (around -5000) in relatively few environment steps in the pretraining environment. However, where the model-based approach truly shines is in the large environment. When compared to the model-free approach, we see that the pretraining in the smaller environment greatly reduces the number of samples required in the larger environment, which is expected

as we saw relatively small errors in the previous section in the models after only one iteration in the new environment. Furthermore, the model-based approaches reach higher rewards than the model-free approach, showing good generalization of the Q-learning approach.

GNN Dynamics Model

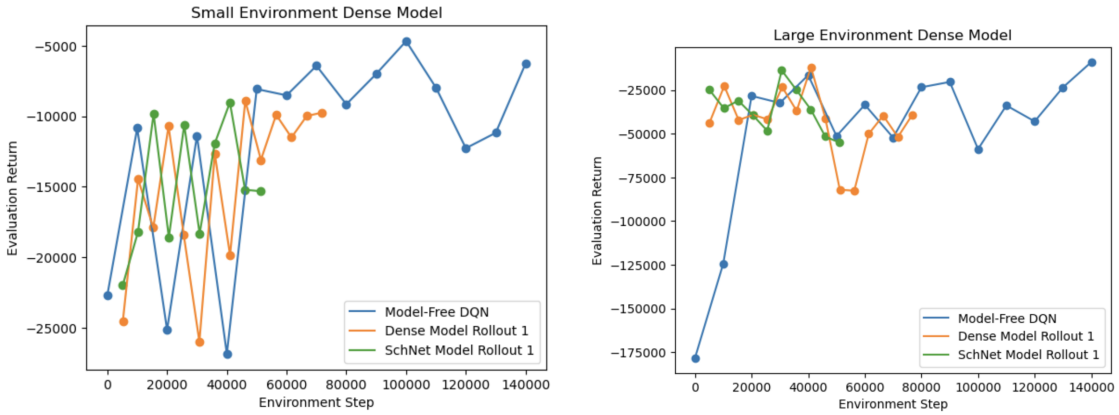

Fig 7. Pretraining and New Environment Deployment Step with the SchNet Model

Similar to the previous section, I include the pretraining evaluation curves and deployment under a distribution shift in the larger environment for the SchNet model-based approach. The data included with the SchNet model corresponds to around 11 hours of training time. This is dominated by the time required to translate the strain and layer count in the action space to atomic structures that the GNN works over. Batching these data and transferring them to and from the GPU was an expensive overhead than severely decreased the viable experiments that could be run with the SchNet architecture due to large training times. A rollout of one was selected as it was previously observed that the SchNet architecture generally had larger error in reward prediction when compared to the dense network, and the larger rollout with the dense architecture corresponded to relatively noisy training.

While the small-environment pretraining doesn't show significantly lower sample complexity to reach high evaluation returns when compared to the model-free approach, we again see a significantly higher evaluation return in the larger environment of the pretrained models, or equivalently a lower sample complexity to reach the same returns. SchNet shows higher rewards than the dense network in the new environment. This might be the case because while the SchNet had on average similar to lower accuracy in model dynamics in the new environment as we saw, it might have been more robust across the domain (max error might have been lower) since it contains fewer parameters (won't overfit) and is based on physical intuition (has a functional form closer to the true model dynamics). This may have led to a reduced max error, which is important for Q-learning since we maximize over the action space. SchNet also shows lower error than the model-free DQN, which had access to the true system dynamics (at great computational cost), showing that a warm start is valuable for lower sample requirements. This shows that our hypothesis well-motivated, and that pretraining a model-based system (notably one with appropriate inductive biases) for inverse material design is an effective way to reduce sample complexity in novel chemical domains.

Addendum

Out of interest, I included an experiment with the three architectures without training the model on the samples in the new environment. This was to see if the model generalized well enough to inform the search in the larger chemical environment. While the dense model performed relatively well, SchNet had very high errors that led to destabilized training, again contrary to our expectations. This may be because the system change with larger layer-count is significant due to the rapid change in local chemical environment. SchNet and the dense model responded differently (SchNet predicted very large DOS, while the dense model, only seeing strain and number of layers from the minimum, predicted similar DOSes from the smaller environment, which weren't too far off).

References

[1] E. Pan et al. "Deep Reinforcement Learning for Inverse Inorganic Materials Design". NeurIPS AI4Mat Workshop (2022)
[2] M. Popova et al. "Deep reinforcement learning for de novo drug design". Science advances (2018)
[3] Z. Zhou et al. "Optimization of molecules via deep reinforcement learning". Scientific reports (2019)
[4] M. Olivecrona et al. "Molecular de-novo design through deep reinforcement learning". Journal of Cheminformatics (2017)
[5] Y. Wu et al. "Spatial Graph Attention and Curiosity-driven Policy for Antiviral Drug Discovery". (2021)
[6] P. Polishchuk, "CReM: chemically reasonable mutations framework for structure generation". Journal of Cheminformatics (2020)
[7] F. Sui, "Deep Reinforcement Learning for Digital Materials Design". ACS Materials Letters (2021)
[8] M. Balog et al "DeepCoder: Learning to Write Programs". ICLR (2017)
[9] Jancu, Scholz: Empirical spds* tight-binding calculation for cubic semiconductors: General method and material parameters
[10] Brahma, Bhattaram, "Accelerated Modelling of Interfaces for Electronic Devices using Graph Neural Networks", NeurIPS AI4Mat Workshop (2023)
[11] Scutt et al. "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions". NeurIPS 2017
[12] Merchant, A., Batzner, S., Schoenholz, S.S. et al. Scaling deep learning for materials discovery. Nature 624, 80–85 (2023).
[13] Nearsightedness of Electronic Matter, Prodan and Kohn, https://arxiv.org/pdf/cond-mat/0503124.pdf