Introduction

What is electron density?
Electron density is a measure of the probability of an electron being present at a specific location. Knowing the electron density gives us information on the bonding of atoms, which give rise to advances in drug discovery and batteries. Current methods for calculating electron density are expensive, with the best methods using density functional theory (DFT) being $O(N^3)$.

How can machine learning (ML) help?
Given a set of input output pairs, a ML algorithm can learn to produce output that resembles output from the original data set distribution. ML holds the promise for bypassing the need to do expensive DFT computations.

What is the current state of the art?
- Brockherde and others obtained electron density by learning Hohenberg Kohn mapping from gaussian potentials to electron density.
- Their method used a kernel ridge regression (KRR) model for learning. The accuracy of KRR modes is sensitive to the selection of training data. For KRR method, there is need for a large number of weights in order to capture nonlinearity.
- Related: Isola and others presented idea of “given an outline, fill it in” in their paper Image-to-Image Translation with Conditional Adversarial Networks.
- Deep learning methods have been successful with capturing nonlinearity, with the most popular deep learning architectures being convolutional neural networks.

Methods
Generate the dataset
We use ab initio methods to calculate the electron density for 625 different configurations of a Lithium-Oxygen-Lithium system. These electron densities are our labels.

From the same calculations, we look at the atoms and their positions, and model the electron density for each atom using gaussians. The resulting matrix of electron densities are our inputs.

Choose a model
Inspired by pix2pix, we choose a U-net architecture. We use L2 norm loss function and batch normalization layers.

Consists of a contracting path and an upsampling path. Every layer but the last performs a series of convolutions, batch norm, and then a non linear activation function called Rectified Linear Unit (ReLU).

We use Unet implementation by Aker et al.

Run and evaluate experiments
Inputs (electron density modeled as gaussians) are fed into a mixer, which then feeds into the model for training. Each input has a corresponding label. After every training iteration, the model makes a prediction and compares the prediction against the label. Weights of the model are updated in order to reduce the loss.

After training is finished we evaluate our model against a test set with electron densities the model did not encounter during training. We calculate a average loss over all points in the test set. The goal of hyperparameter tuning is to reduce this average loss.

Results
Our models generate predictions that are very close to our labels. Many predictions, however, come with some extra “dust” of electron density in between atoms.

Additionally, as evidenced by filters that our model outputs, the model learns to differentiate between lithium and oxygen atoms without being told what the atoms are. That is, the model appears to be learning the underlying chemistry of our system.

As a result of hyperparameter tuning using a parallelized grid search, we realized that dropping dropout led to increased performance on the unet architecture. This observation supports existing literature, which recommends the removal of dropout when using batch normalization.

Conclusions
Future research endeavors will be focused towards the application of tools from optimal transport.

Long term goals are to see if a model trained on multiple sets of configurations of atoms can correctly predict electron density for a new and larger system that the model has never encountered before.

We would like to acknowledge the support of the NSF through the grant ACI-1560168.

References
1. Wikipedia
2. Understanding the Disadvantages of Gradient Descent by Yousef Shabtai and others
3. Image to Image Translation with Conditional Adversarial Networks by Philipp Isola and others
4. Related: Isola and others presented idea of “given an outline, fill it in” in their paper Image-to-Image Translation with Conditional Adversarial Networks.
5. Deep learning methods have been successful with capturing nonlinearity, with the most popular deep learning architectures being convolutional neural networks.