CS236 Suggested Project Ideas

1. **Progressive Flow Models**
    - **Project Description:** Flow models are appealing due to its reversible structure, allowing the users to compute tractable likelihoods. Similar to Progressive GANs / PixelCNNs, one could also think of means learn a flow network that allows Images to grow progressively in size. This could be possible by means of the checkerboard partition over the images -- one could either add Gaussian noise at low resolution levels, or consider a "deconvolution" structure that first transforms low resolution images to high resolution ones, and then "fine-tune" the high-resolution ones.
    - **Contact:** Jiaming Song (jiaming.tsong@gmail.com)
2. **Algorithmic Regularization for Deep Generative Models**
    - **Project Description:** Regularization clearly matters in supervised learning. Even with the same architecture, the algorithm can drastically affect performance, possibly due to better learning rate schedules (https://arxiv.org/abs/1608.03983) and data augmentation (mixup). However, there are relatively fewer discussion about this on deep generative models, as most focus has be over the architecture and loss functions. Consider extending the regularization techniques used in (https://arxiv.org/abs/1805.08913) , and see how does the regularization techniques in supervised deep learning translate to deep generative models.
    - **Contact:** Jiaming Song (jiaming.tsong@gmail.com)
3. **Sinkhorn GAN:**
    - **Project Description:** Wasserstein GAN (https://arxiv.org/abs/1701.07875) uses the Kantorovich-Rubinstein dual formulation of the Wasserstein distance, which requires the critic to be Lipschitz. To satisfy this constraint, many regularization techniques have been proposed for Wasserstein GAN, for example, weight clipping and gradient penalty (https://arxiv.org/abs/1704.00028). In contrast, the dual problem of Sinkhorn distance does not require Lipschitz smoothness (https://arxiv.org/abs/1605.08527), which might be a good way to get rid of regularizations. Try to use neural networks to solve the dual of Sinkhorn distance and see whether it improves over Wasserstein GAN with weight clipping or gradient penalty.
    - **Contact:** Yang Song (songyang@stanford.edu)
4. **Generating Point Clouds:**
    - **Project Description:** A point cloud can be viewed as the collection of iid samples from a distribution. Modeling point clouds can thus be viewed as learning family of distributions. Try to build a VAE to generate point clouds. The encoder can be based on DeepSet (https://arxiv.org/abs/1703.06114). See a

starting example in section 4.5 of https://arxiv.org/pdf/1801.09819.pdf. Also https://openreview.net/pdf?id=BJInEZsTb might be useful.
    - **Contact:** Yang Song (songyang@stanford.edu)
5. **Parallel Auto-Regressive Image Flows:**
    - **Project Description:** Apply Probability Density Distillation (https://arxiv.org/pdf/1711.10433.pdf) to image domain. The goal is to try to drastically improve the speed of sampling for auto-regressive models.
    - **Contact:** Rui Shu (ruishu@stanford.edu)
6. **An Empirical Study of VAE Image Quality**
    - **Project Description:** It is surprisingly challenging to get VAEs to generate high-quality samples of complicated, high-dimensional continuous image distributions (much in contrast to GANs). This will be an empirical study to see what kinds of VAE models are capable of modeling continuous data, to develop a set of heuristics to guide future VAE practitioners, and/or to open-source your implementations of what worked.
    - **Contact:** Rui Shu (ruishu@stanford.edu)
7. **Data Augmentation using GANs**
    - **Project Description:** Choose an image classification problem that is interesting to you. Train a conditional GAN to produce new images of a given label, and train a classifier using the synthetic data in addition to the original data. How does it compare to a classifier trained using just the original data? Does the amount of synthetic data you use make a difference?
    - **Contact:** Casey Chu (caseychu@stanford.edu)
8. **Fun GAN Applications**
    - **Project Description:** There are lots of fun things that GANs can do (e.g. style transfer, text-to-image, super-resolution). Check out this page (https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900) for inspiration. Reproduce a GAN-related project and propose an extension, perhaps applying it to different datasets and/or combining multiple techniques.
    - **Contact:** Casey Chu (caseychu@stanford.edu)
9. **VAE Training with Optimal Decoder**
    - **Project Description:** It is possible (albeit costly) to compute the optimal decoder of a VAE. What if we compute (or approximate) the optimal decoder every once in a while during training and directly train the VAE decoder using the optimal decoder? How would that change the learning landscape? Would it give sharper images?
    - **Contact:** Rui Shu (ruishu@stanford.edu)
10. **Efficient Image Compression with GANs**
    - **Project Description:** GANs have been shown to be able to compress images with extremely low bitrates. However, there is no open-source implementation or study of the efficiency of these methods. In this project,

you would explore methods of compressing images with GANs and building efficient models for image compression.
  - **Contact:** Daniel Kang (ddkang@stanford.edu)

11. Cycle Consistency for Normalizing Flow Models
  - **Project Description:** CycleGAN allows for translating between two different domains of images in an unsupervised way (https://junyanz.github.io/CycleGAN/). It does so by augmenting the GAN loss with another term that ensures that translating from one domain to another is reversible. Normalizing flow models automatically guarantee such consistency and can be trained adversarially, using MLE or even a hybrid (https://arxiv.org/abs/1705.08868). This project will aim towards image to image translation via a flow model trained adversarially.
  - **Contact:** Aditya Grover (adityag@cs.stanford.edu)

12. Molecule Generation via Iterative VAEs
  - **Project Description:** Molecules can be visualized as graphs with atoms as nodes and bonds between atoms as edges. Unlike images, graphs are discrete and combinatorial in size. Hence, molecule generation via a generative model is a challenging task. Limited success is shown on generation of small molecules (https://arxiv.org/pdf/1802.03480.pdf). On the other hand, another recent work Graphite (https://arxiv.org/abs/1803.10459) proposes an iterative refinement procedure combined with graph neural networks that can scale to large graphs for density estimation and other inference tasks. The goal of this project is to combine the various advancements in VAEs (some of which you'll implement in HW2) with the open sourced implementation of Graphite to generate molecules and materials.
  - **Contact:** Aditya Grover (adityag@cs.stanford.edu)