Project Review: *Mixture GAN* by Tikhon Mavrin

Reviewer: Hassan Iftikhar

5/13/2025

1. Does the Report Include the Required Parts?

- **Problem Statement:** Yes. The paper clearly explains that standard GANs suffer from a vanishing gradient problem when the real and fake distributions do not overlap. This causes the discriminator to become too confident and stops the generator from learning.
- **Main Idea:** Yes. The author proposes training the GAN on mixtures of real and fake samples using a mixing factor $t \in [0, 1]$. This is done to maintain gradient flow and improve training stability.
- **Comparison with Relevant Methods:** Yes, but very short. Methods like WGAN, WGAN-GP, spectral normalization, and non-saturating loss are mentioned, but no experiments or comparisons are provided.

2. Clarity and Missing Information

- The idea is generally understandable, but the math is written in a rough way, and variables like x_t, x_d, x_g, and $\mu[0, 1]$ are not properly defined in this equation 4 in the paper.
- The baseline section lacks detail: it does not say what data was used but after seeing the code we know swiss-roll dataset was generated for this purpose.
- There are no results or plots in the PDF report even though they are available in the code.
- **Suggestions for improvement:**
  - Add labeled figures to explain the method.
  - Include training results and plots.
  - Clean up and define all math terms.
  - Summarize comparison with existing methods in a small table.

3. Styling, Quality, and Structure

The paper is mostly well-organized and easy to follow. Styling is fine. Adding one or two figures (such as model architecture or the mixing diagram) would make the explanation much better.

4. Experiment Protocol

Yes, the experiment setup is correct. The model is trained using a valid sampling method with the time variable t. The generator and discriminator update steps are correct, and logging is done properly. The protocol makes sense and is complete.

5. Presented Results

Yes, the results shown in the Jupyter notebook are reasonable. The generator improves over time and learns to mimic the true distribution. However, these plots should be included in the PDF report. A few more training examples or metrics would make the evaluation stronger.

6. GitHub Structure

The code is placed inside a single notebook and is cleanly written. Maybe, it would be better to split the code into separate Python files in future.

7. README File

There is no `README.md` file.

8. Reproducibility

The code is mostly reproducible. However:

- It starts with `device = "cuda:6"`, which may not work on other machines.
- The user needs to manually enter a Weights & Biases API key to log experiments.

Mentioning these in the instructions would help.

9. Code Execution

Yes, the code runs without errors after setting up the environment properly. No bugs were found during training.

10. Final Suggestions for Improvement

- Add a short and clean `README.md`.
- Include training plots and results in the main PDF report.
- Define math variables more clearly.
- Add simple figures for explanation.
- Compare results with other GAN variants to show strengths.

Conclusion

The code works well and shows promising results. With some small improvements in the report structure, math clarity, and documentation, this could be a very strong project.