Cornell University

CS 5788: Introduction to Generative Models

Spring 2026   Instructor: Andrew Owens

Project Proposal

Posted: Tuesday, February 24, 2026 **Due:** Thursday, March 19, 2026

Please submit your written proposal to Gradescope as a `.pdf` file.

Deliverable: Fill in the project proposal template. Make sure to replace all instructions (in italics) with your own responses. **Page limit is 2 pages**. References section do not count towards the page limit. If you prefer, you may use a text editor other than LaTeX, but it should closely match the style of the template (0.8 inch page margins, Times New Roman font, font size 12 for the main text).

Open source policy. Your project must demonstrate significant technical contribution of your own. You are permitted to use open-source libraries and models as **building blocks**, but you **may not submit any existing repositories** as your primary deliverable. You also **may not use high-level API wrappers** that abstract away the core process. Any open-source code used must be substantially modified, extended, or combined in a novel way to address your problem statement. For example:

- **Permitted:** Using an open-source image encoder (e.g., ResNet) and a text decoder (e.g., GPT-2) to build a new image captioning system.
- **Not permitted:** Downloading an existing image captioning repository and submitting it as your project.
- **Not permitted:** Using an API wrapper (e.g., `StableDiffusionInpaintPipeline` or `AudioLDMPipeline`) to perform the core task with a single function call.

Topics. Select a topic from the list below or propose your own. Topics may be changed later by submitting a new proposal to the staff email (cs5788-staff-2026sp-L@cornell.edu). **You may begin working on your project right away**. We will manually review and approve the proposals, but we rarely reject topics outright.

We will be continuously updating this list of suggested project topics (last update: Feb. 24).

- **Training energy-based models.**
  - Compare different methods for training energy-based models.
  - Related work:
    - [1] How to Train Your Energy-Based Models
    - [2] Improved Contrastive Divergence Training of Energy Based Models

- **Watermarking generative models.**
  - Implement an invisible watermarking strategy for either language or image generation model that operates during the generation process (e.g., modifying logits for LLMs or latent noise for images). *Do not use post-processing techniques like LSB (Least Significant Bit) replacement.*
  - Show that your watermarking technique is robust.
    * For language: show they are detectable under various lengths (e.g., page, paragraph, sentence).
    * For image: show they are detectable under various degradations (e.g., blur, resize, JPEG compression).
  - Related work:
    * Language:
      · [1] A Watermark for Large Language Models
      · [2] Scalable watermarking for identifying large language model outputs.
    * Vision:
      · [1] Tree-Ring Watermarks: Fingerprints for Diffusion Images that are Invisible and Robust
      · [2] HiDDeN: Hiding Data With Deep Networks
- **Analyzing diffusion language models.**
  - Diffusion language models generate text all-at-once, whereas models such as GPT generate text autoregressively (left-to-right).
  - Identify and implement at least two tasks where this architectural difference significantly impact performance (e.g., accuracy, model behavior, speed). Quantitatively compare pretrained diffusion models against autoregressive baselines and analyze the observed differences in detail. Your analysis should **include at least two models for each type.**
  - *Note: you may start from pretrained models for this project.*
  - Related work:
    * [1] Diffusion-LM Improves Controllable Text Generation
    * [2] Simple and Effective Masked Diffusion Language Models
- **Exploring the Platonic Representation Hypothesis.**
  - Huh *et al.* argue that as AI models scale, their representations converge toward a shared statistical reality ("The Platonic Representation"), regardless of architecture or modality.
  - Empirically verify this phenomenon by analyzing the internal representations of **3 or more architecturally distinct** generative models.
  - *Note: you may start from pretrained models for this project.*
  - Related work:
    * [1] The Platonic Representation Hypothesis
    * [2] Do Vision Transformers See Like Convolutional Neural Networks?
- **Real image editing.**

- Implement real image editing using generative models. Show that your method works on real images of your own (e.g., photos taken using your camera) and accurately follows your text prompt.
- Related work:
  * [1] Prompt-to-Prompt Image Editing with Cross-Attention Control
  * [2] Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation

- **Generative audio editing.**
  - Implement a text-to-audio generator that can edit a given audio (e.g., inpainting). Show that your model can edit real-life audio or music, and correctly follows the given text prompt.
  - Related work:
    * [1] AudioLDM: Text-to-Audio Generation with Latent Diffusion Models
    * [2] Simple and Controllable Music Generation

- **Multimodal models.**
  - Take existing models for different modalities (e.g., vision, language, audio) and architecturally combine them to create a multimodal model.
    * You may look into smaller models such as nanochat.
  - Demonstrate the capabilities of your multimodal model (e.g., multimodal understanding or generation).
  - Related work:
    * [1] Visual Instruction Tuning
    * [2] Qwen-Audio:Advancing Universal Audio Understanding via Unified Large-Scale Audio-Language Models

- **Faster diffusion using few-step distillation.**
  - Implement a few-step distillation pipeline for speeding up the diffusion process. Compare the generation quality and speed between the original model and your distilled model.
  - Related work:
    * [1] Progressive Distillation for Fast Sampling of Diffusion Models
    * [2] Adversarial Diffusion Distillation

- **Consistent character generation.**
  - Fine-tune an existing diffusion model to consistently generate a character of your choosing.
  - Implement and compare at least two different personalization methods and analyze the consistency across various generation settings.
  - Related work:
    * [1] DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation
    * [2] An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion

- **Language models and RLHF (Reinforcement Learning from Human Feedback).**
  - Implement and compare at least two different RLHF methods that modifies the behavior of the language model.
  - *Note: you may start from a pretrained language model for this project. However, the RLHF implementation must be your own.*
  - Related work:
    * [1] WebGPT: Browser-assisted question-answering with human feedback
    * [2] Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Proposing your own project direction: If you wish to propose a project direction that does not cleanly fall into one of these suggested topics, the topic should be extremely relevant to generative modeling. Types of project include, but not limited to: implementing a novel algorithm, combining several existing methods, or analyzing an interesting problem.