Homework 3: Baseline Implementation & Research Foundation

CMU 10-799: Diffusion & Flow Matching

Spring 2026

Due: Sunday, February 15, 2026 at 11:59 PM ET **Total Points:** 100
Late Due Date: Tuesday, February 17, 2026 at 11:59 PM ET
Submission: Gradescope https://www.gradescope.com/courses/1207241
Starter Code: https://github.com/KellyYutongHe/cmu-10799-diffusion/

Introduction

Welcome to Homework 3! In HW1 and HW2, you built foundational generative models (DDPM, Flow Matching, DDIM) and chose your specialization track. Now it's time to dive deep and build your own path.

In this homework, you'll research your chosen problem space, select an existing method as your baseline, and implement it. Think of this as laying the groundwork for a mini research project: you're choosing an interesting problem, doing the literature review, background research, and establishing a baseline that you'll improve upon in HW4.

By the end of this homework, you'll have:

1. A clearly defined problem with specific assumptions and evaluation criteria
2. A survey of existing approaches in your area
3. A working implementation of a baseline method
4. Preliminary results and ideas for improvement

Special note for HW 3 & 4:

1. **This homework is AI-friendly.** You may use any AI coding assistants, chatbots, or reference implementations. You may also use any other resources that you can find on the Internet. You may also use AI assistants for writing. However, you are responsible for every single word you submit. In other words, if something is wrong, unclear, or plagiarized, it's on you. AI is a tool, not an excuse. At the end of the homework, you'll document what resources you used.
2. **Build on your HW1&2 codebase.** Your implementation should extend from the codebase you've been developing. If you find useful code online (e.g., a reference implementation of your baseline method), you may incorporate it into your codebase directly. Just cite your sources clearly and explain what you used and modified.
3. **You must use the same dataset.** You may **not** switch to a different dataset (e.g., your own protein dataset) or use a different resolution for the last two homework. You may use additional training data or pretrained models if your method requires it, but all evaluation metrics (KID, FID, etc.) must be computed on the same CelebA 64×64 subset from HW1 and HW2. This ensures fair comparison across homeworks and across students.

Part I: Motivation & Problem Definition (20 points)

Now let's start doing your mini research project! In the first part, you will be thinking about and writing what corresponds to the **Introduction** and **Problem Statement** sections of a research paper. In the introduction, you motivate why your problem matters and what gap you're addressing. In the problem statement, you formally define the task, inputs, outputs, and evaluation criteria.

Q1. Track Selection [1 pts]

Which track did you choose? (Fidelity / Controllability / Speed)

Note: You may change your track from what you submitted in HW2, but after submitting this homework, your track is locked for HW4.

Q2. Motivation [9 pts]

(a) [3 pts] Why did you choose this track? What personally interests you about this challenge?

(b) [3 pts] What is the core challenge? In 1-2 paragraphs, describe the fundamental limitation or problem within your track that you want to address.

(c) [3 pts] Why does this matter? Give 2-3 concrete applications or scenarios where solving this problem would be valuable. Be specific and not just "it would make diffusion models better."

Q3. Problem Definition [10 pts]

(a) [4 pts] What specific problem are you solving? Your track is broad, let's narrow it down. In 1-2 paragraphs, describe the specific subproblem you'll focus on and why it is interesting. Some ideas within each track:

- **Fidelity:** Better architectures, improved noise schedules, loss objective modifications, etc.
- **Controllability:** Attribute-conditional generation, text-to-image generation, image-to-image generation, image editing, inverse problem solving, style transfer, spatial control, etc.
- **Speed:** 1-step generation, few-step generation, better ODE solvers, adaptive step sizes, distillation, flow maps, etc.

(b) [4 pts] What are your inputs, outputs, and assumptions?

- **Input:** What does your method take in? Be specific about format, dimensions, and any conditioning information.
- **Outputs:** What does it produce? Be specific about format, dimensions, and any other information.
- **Assumptions:** What are you assuming or what do you have access to? (e.g., pretrained models, labels, external data, frozen components)

(c) [2 pts] How will you measure success? What are your primary metrics? What would a "successful sample" from your model look like?

Part II: Related Works (20 points)

Time to see what's already out there! This part corresponds to the **Related Work** section of a research paper. Here you'll survey existing approaches, compare them, and understand where your work fits in the landscape. A good related work section shows you understand the field and helps you identify what's missing, i.e. the gap your work will address!

Q4. Literature Survey [20 pts]

(a) [12 pts] Survey 4-6 relevant methods for your specific subproblem. For each method, include:

- **Full citation**: Include authors, title, venue, year. You can use LaTeX bibliography for this.
- **2-3 sentence summary:** What's the key idea?
- **Pros and cons:** What does it do well? What are its limitations?

(b) [8 pts] Include a comparison table below. Compare the methods you surveyed along 3-4 relevant dimensions. Choose dimensions that matter for your problem.

Here are some examples of comparison tables from existing papers:

Table 1: A comparison of several common paradigms for generative modeling. [Explicit $\mathbf{x} \to \mathbf{z}$]: the mapping from $\mathbf{x}$ to $\mathbf{z}$ is directly trainable, which enables SSL; [No prior hole]: latent distributions used for generation and training are identical (Sec. 4.2), which improves generation; [Non-adversarial]: training procedure does not involve adversarial optimization, which improves training stability.

Model family	Explicit $\mathbf{x} \to \mathbf{z}$ (Enables SSL)	No prior hole (Better generation)	Non-Adversarial (Stable training)
VAE [49, 74], NF [26]	✔	✗	✔
GAN [32]	✗	✔	✗
BiGAN [28, 30]	✔	✔	✗
DDIM [81]	✗	✔	✔
D2C	✔	✔	✔

Figure 1: The related work comparison table in Sinha et al. [2021].

Table 1: A conceptual comparison between our proposed LADiBI and the existing literature.

Method Family	Method	Prior Type	Diverse Image Prior	Training-free	Flexible Operator
Optimization-based	Pan-ℓ_0 [Pan et al., 2017]	Inductive bias	×	✓	×
	Pan-DCP [Pan et al., 2018]	Inductive bias	×	✓	×
Self-supervised	SelfDeblur [Ren et al., 2020]	Inductive bias	×	✓	×
Supervised	MPRNet [Zamir et al., 2021]	Discriminative	×	×	✓
	DeblurGANv2 [Kupyn et al., 2019]	GAN	×	×	✓
Diffusion-based	BlindDPS [Chung et al., 2023a]	Pixel diffusion	×	×	×
	BIRD [Chihaoui et al., 2024]	Pixel diffusion	×	✓	×
	GibbsDDRM [Murata et al., 2023]	Pixel diffusion	×	✓	×
	LADiBI (Ours)	Text-to-image latent diffusion	✓	✓	✓

Figure 2: The related work comparison table in Dontas et al. [2024].

Part III: Baseline Selection & Preliminaries (25 points)

Now it's decision time! This part reflects a critical research decision: choosing what to build on. In a research paper, though usually done implicitly, you would generally compare against certain selection of baselines and build on top of a few methods. Here, we're asking you to make that decision explicit and justify it.

Then, before diving into implementation, let's also make sure you really understand your chosen method. This part corresponds to the **Background** or **Preliminaries** section of a research paper. Here you'll explain the technical foundation needed to understand your approach. After reading this section, a classmate should be able to understand how your baseline work, even if they've never heard of it before.

Q5. Choose Your Baseline [7 pts]

(a) [3 pts] Which method will you implement? State clearly which paper/technique you're implementing as your baseline.

(b) [3 pts] Why this method? Why did you choose this over the others you surveyed?

(c) [1 pts] What resources will you use? (reference code, pretrained models, tutorials, etc.)

Q6. Technical Deep Dive [18 pts]

Now let's explain your baseline method in enough detail so that a classmate could easily implement it by just looking at your answers to the following questions.

(a) [5 pts] What is the key insight? In 1-2 paragraphs, explain the core idea behind the method. What makes it work? Include a diagram if it helps clarify the idea.

(b) [5 pts] How does the algorithm work? Provide pseudocode or a step-by-step description of the method.

(c) [5 pts] What is the key equation? Write out at least one important equation and explain what each term means.

(d) [3 pts] What are the important hyperparameters or design choices? What values does the original paper use, and what values will you use?

Part IV: Implementation & Preliminary Results (25 points)

Time to get your hands dirty! This part corresponds to contents in the **Experiments** section of a research paper, specifically, the implementation details and preliminary baseline results. You'll describe what you built, verify it works, and show some initial outputs. The goal here isn't to beat state-of-the-art – it's to establish that your baseline is correctly implemented and producing reasonable results.

Q7. Experimental Setup [10 pts]

(a) [8 pts] Describe your experimental setup:

- Model architecture and size (number of parameters)
- If your method requires training: batch size, learning rate, iterations, optimizer, etc
- Inference hyperparameters: number of sampling steps, solver, sampling noise schedule, guidance scale, etc
- If applicable: what pretrained model are you using? what external dataset are you using?
- If your method has task-specific setup: describe it (e.g., degradation type/parameters for inverse problems, which attributes for conditional generation, distillation schedule for speed methods)
- Any modifications to the original paper

(b) [2 pts] How much compute did you use? (GPU type, training/inference time)

Q8. Preliminary Results [15 pts]

(a) [10 pts] Show preliminary qualitative results from your baseline. That is, show a grid of samples (or equivalent for your track—e.g., before/after for editing, conditional samples for controllability, samples at different step counts for speed) below.

(b) [(optional) 0 pts] If you already have quantitative results, report metrics (KID, accuracy, etc.) here. If not, explain what you plan to measure in HW4.

(c) [5 pts] Based on your preliminary results: What's working well? What needs improvement? Any surprises or unexpected behaviors?

Part V: Brainstorming for HW4 (5 points)

You've got a working baseline—now let's think ahead! This part is a bridge to HW4, where you'll propose and implement improvements. In a research paper, this would seed your **Methods** section – the novel contribution you're making beyond existing work. For now, you're just brainstorming what that contribution might be.

Q9. Ideas for Improvement **[5 pts]**

(a) [5 pts] Brainstorm 2-3 ideas for how you might improve upon your baseline in HW4. For each: What would you change or try? Why do you think it might help?

These don't need to be fully fleshed out. They are just seeds for HW4. Remember: simple hyperparameter sweeps don't count as innovation. We want to see you try something new.

Part VI: Reflection (5 points)

Q10. Reflection **[5 pts]**

(a) [2 pts] What was the most interesting or surprising thing you learned from your literature survey?

(b) [2 pts] Did implementing the baseline change your understanding of the method? How?

(c) [1 pts] List all the resources you used for this homework: AI tools, open source code, tutorials, papers, classmates, etc.

That's it! You have completed HW 3! You've done the hard part, i.e. understanding the problem and getting a baseline working. Now the fun begins in HW4!

References

Michail Dontas, Yutong He, Naoki Murata, Yuki Mitsufuji, J Zico Kolter, and Ruslan Salakhutdinov. Blind inverse problem solving made easy by text-to-image latent diffusion. *arXiv preprint arXiv:2412.00557*, 2024.

Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2c: Diffusion-denoising models for few-shot conditional generation. *arXiv preprint arXiv:2106.06819*, 2021.