CS492D: Diffusion Models and Their Applications

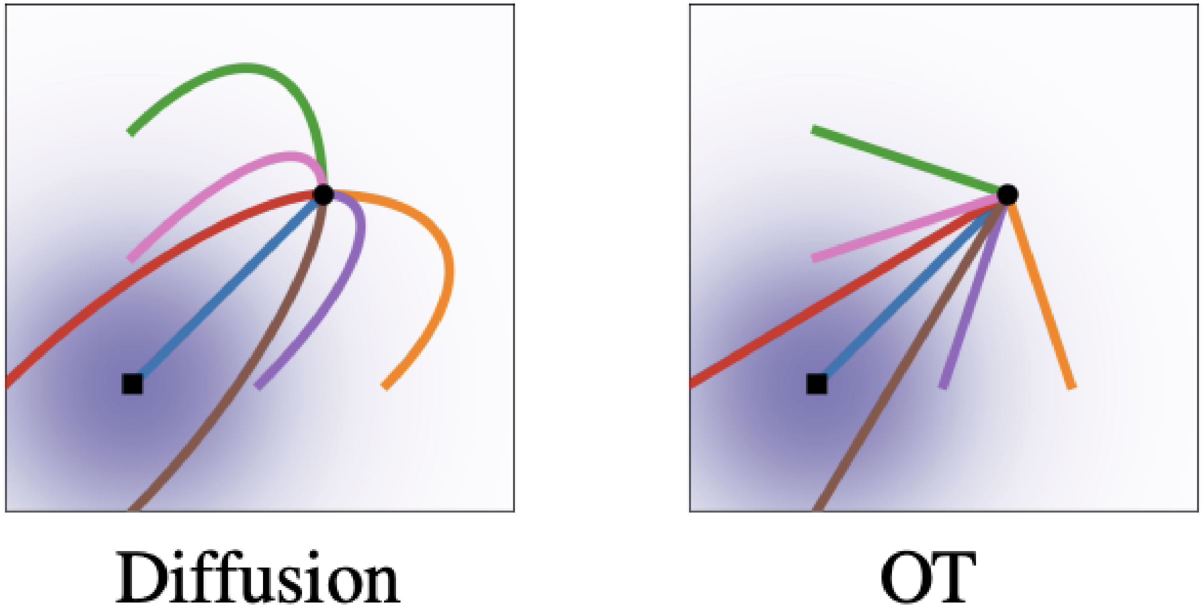
Assignment 7 Session

JUIL KOO

Fall 2024
KAIST

Introduction

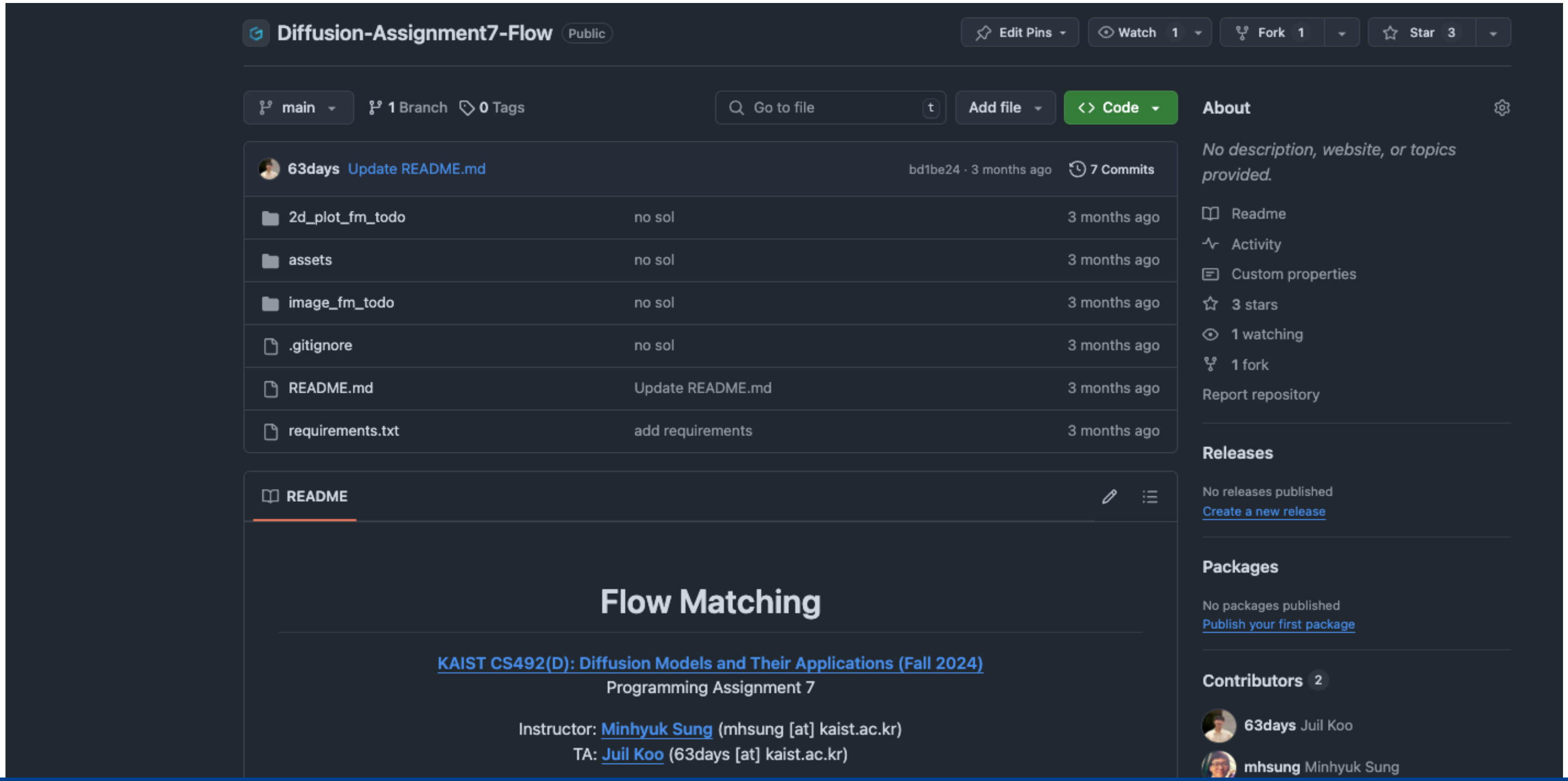
In Assignment 7, we will implement **Flow Matching**, a novel generative model that offers straighter sampling trajectories than diffusion models.



Flow Matching for Generative Modeling, Lipman *et al.*, ICLR 2023.

Introduction

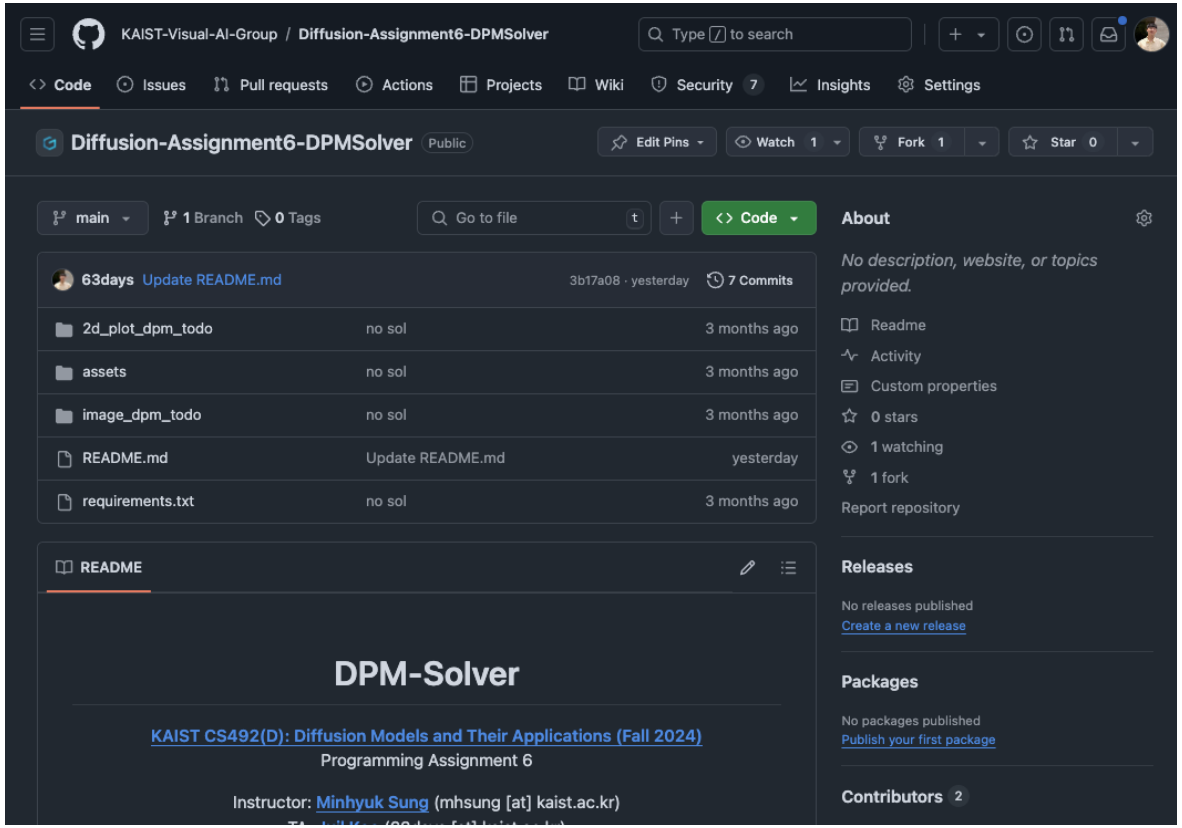
The skeleton code and instructions are available at:
https://github.com/KAIST-Visual-AI-Group/Diffusion-Assignment7-Flow



Introduction

Assignment 7 builds on your implementation of Assignment 2 (DDIM-CFG).

Refer to the `README.md` for details on which files need to be copied.



Important Notes

- Assignment 7 is due on **Dec 13th (Friday)**.
- Late submission will incur **20% penalty** for **each** late day!
- Please carefully check the README of each assignment.
- Missing items in your submission will also incur penalties.

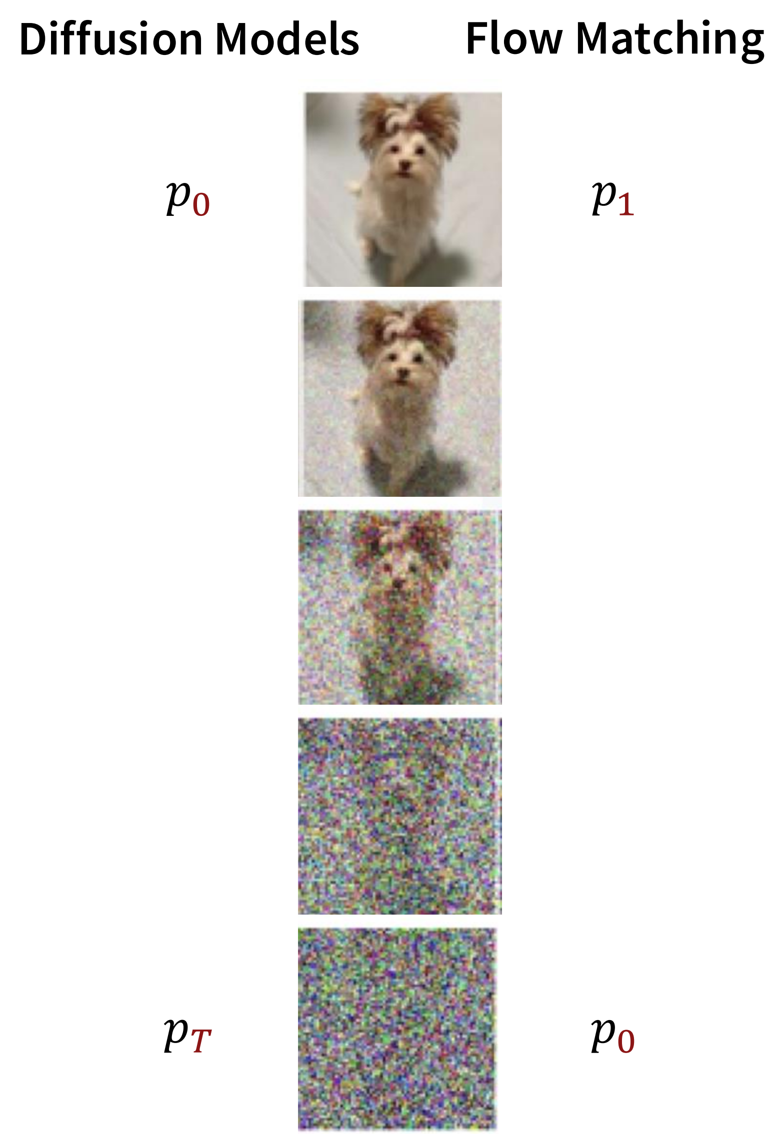
Notations

Diffusion Models [Generation: $T \to 0$]

- $\boldsymbol{x}_0$: A sample from the data distribution.
- $\boldsymbol{x}_T$: A sample from the reference (base) distribution.

Flow Matching [Generation: $0 \to 1$]

- $\boldsymbol{x}_1$: A sample from the data distribution.
- $\boldsymbol{x}_0$: A sample from the reference (base) distribution.



Flow and Vector Fields

- A flow $\phi_t(\boldsymbol{x})$ is a time dependent mapping function, which is similar to the forward pass of diffusion models: $\boldsymbol{x}_t = \phi_t(\boldsymbol{x})$.

- The derivative of $\phi_t(\boldsymbol{x})$ w.r.t t is called a vector field $u_t\big(\phi_t(\boldsymbol{x})\big) = \frac{d\phi_t(\boldsymbol{x})}{dt}$.

Probability Paths

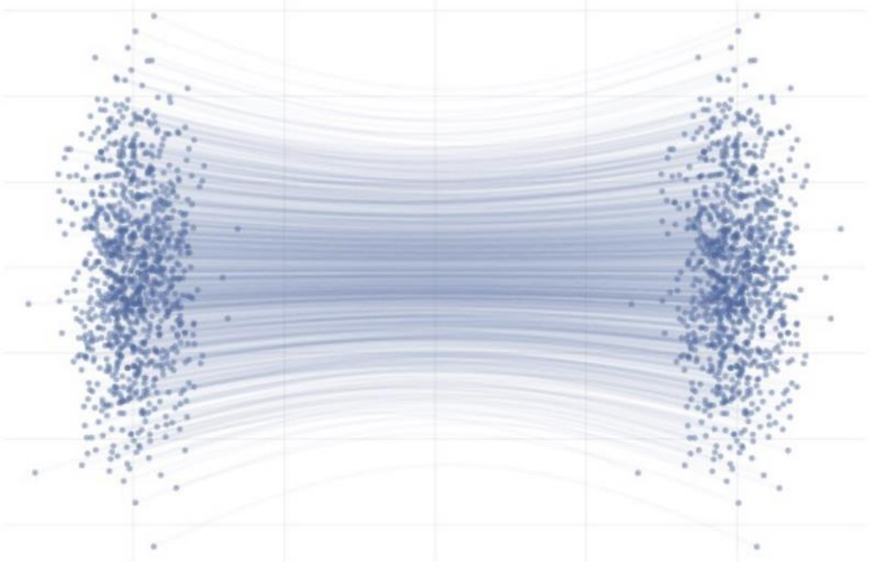
A vector field $u_t(\phi_t(\boldsymbol{x}))$ is said to generate a probability path p_t if its flow $\phi_t(\boldsymbol{x})$ satisfies the equation below:

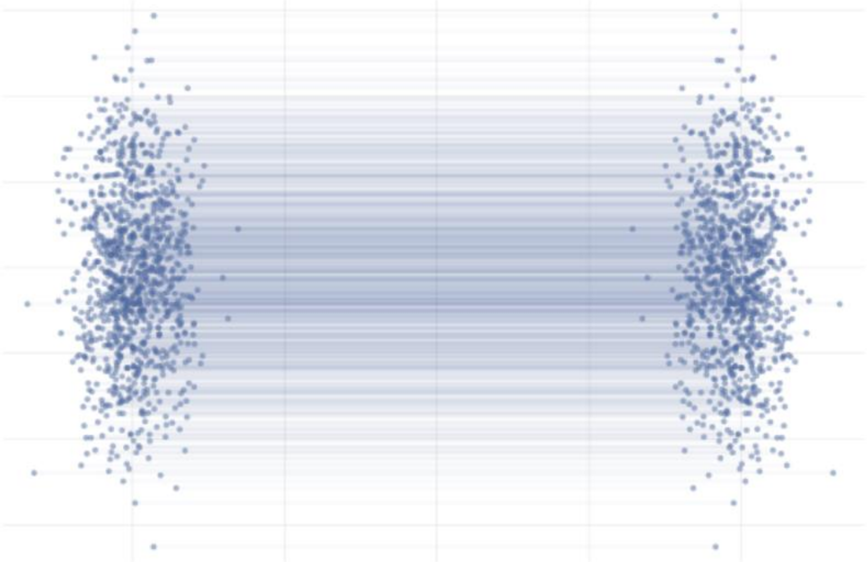
$$p_t(\boldsymbol{x}) = [\phi_t]_* p_0(\boldsymbol{x}) = p_0\left(\phi_t^{-1}(\boldsymbol{x})\right) \det\left[\frac{\partial \phi_t^{-1}}{\partial \boldsymbol{x}}(\boldsymbol{x})\right]$$

Flow Matching

$$\mathcal{L}_{FM} = \mathbb{E}_{t,\,\boldsymbol{x}_t \sim p_t}[\|[v_\theta(\boldsymbol{x}_t, t) - u_t(\boldsymbol{x}_t)\|]^2$$

However, we do not know how p_t and u_t look like. The vector field u_t that generates the probability path p_t is not unique.





https://www.peterholderrieth.com/blog/2023/The-Fokker-Planck-Equation-and-Diffusion-Models/

Conditional Flow Matching

Since the groundtruth vector field $u_t(\boldsymbol{x}_t)$ is unknown,
we define p_t and u_t per sample, i.e., $p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)$ and $u_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)$, where $\boldsymbol{x}_1 \sim p_1$.

$$\mathcal{L}_{FM} = \mathbb{E}_{t,\, \boldsymbol{x}_t \sim p_t}[\|[v_\theta(\boldsymbol{x}_t, t) - u_t(\boldsymbol{x}_t)\|]^2$$

$$\Downarrow$$

$$\mathcal{L}_{CFM} = \mathbb{E}_{t,\, \boldsymbol{x}_t \sim p_{t|1}}\left[\left\|\left[v_\theta(\boldsymbol{x}_t, t) - u_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)\right\|\right]^2\right.$$

Conditional $p_{t|1}$ and $u_{t|1}$

While the conditional probability paths $p_{t|1}$ and vector fields $u_{t|1}$ can be arbitrarily chosen, we adopt the simplest one, which is based on Gaussian distributions:

- **Conditional probability path** $p_{t|1}$:

$$p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1) = \mathcal{N}(\mu_t(\boldsymbol{x}_1), \sigma_t^2(\boldsymbol{x}_1)\boldsymbol{I})$$

- **Conditional flow map** ϕ_t:

$$\phi_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \mu_t(\boldsymbol{x}_1) + \sigma_t(\boldsymbol{x}_1)\boldsymbol{x}_t$$

- **Conditional vector field** u_t:

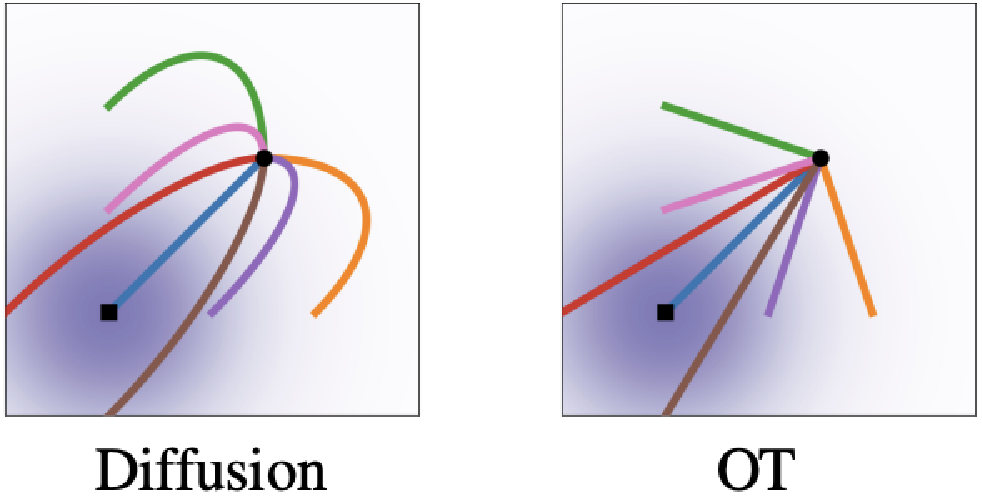
$$u_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \frac{\sigma_t'(\boldsymbol{x}_1)}{\sigma_t(\boldsymbol{x}_1)}\big(\boldsymbol{x}_t - \mu_t(\boldsymbol{x}_1)\big) + \mu_t'(\boldsymbol{x}_1)$$

Special Instance of FM: Straight Path

- Given a flow map $\phi_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \mu_t(\boldsymbol{x}_1) + \sigma_t(\boldsymbol{x}_1)\boldsymbol{x}_t$, we set $\mu_t(\boldsymbol{x}_1) = t\boldsymbol{x}_1$ and $\sigma_t(\boldsymbol{x}_1) = 1 - (1 - \sigma_{min})t$.

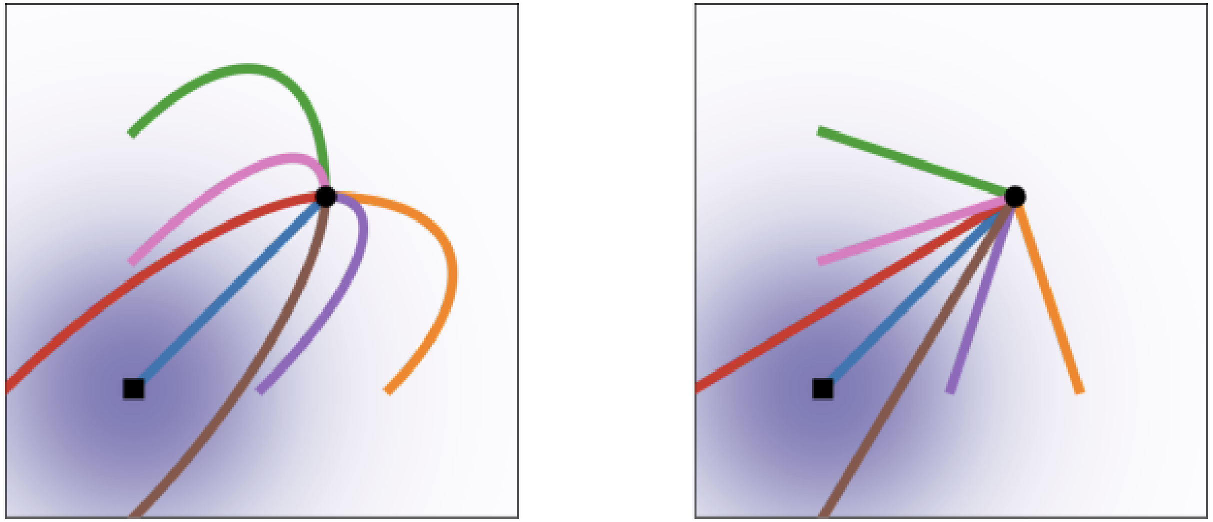
- This results in a simple interpolation between $\boldsymbol{x}_1$ and $\boldsymbol{x}_t$:

$$\phi_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = (1 - (1 - \sigma_{min})t)\boldsymbol{x}_t + t\boldsymbol{x}_1.$$



Special Instance of FM: Straight Path

$$
\begin{aligned}
\mathcal{L}_{CFM} &= \mathbb{E}_{t,\,\boldsymbol{x}_t \sim p_{t|1}}\left[\left\|\left[v_\theta(\boldsymbol{x}_t, t) - u_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)\right\|^2\right]\right. \\
&= \mathbb{E}_{t,\boldsymbol{x}_1,\boldsymbol{x}_0}[\|v_\theta(\boldsymbol{x}_t, t) - (\boldsymbol{x}_1 - (1-\sigma_{min})\boldsymbol{x}_0)\|^2]
\end{aligned}
$$



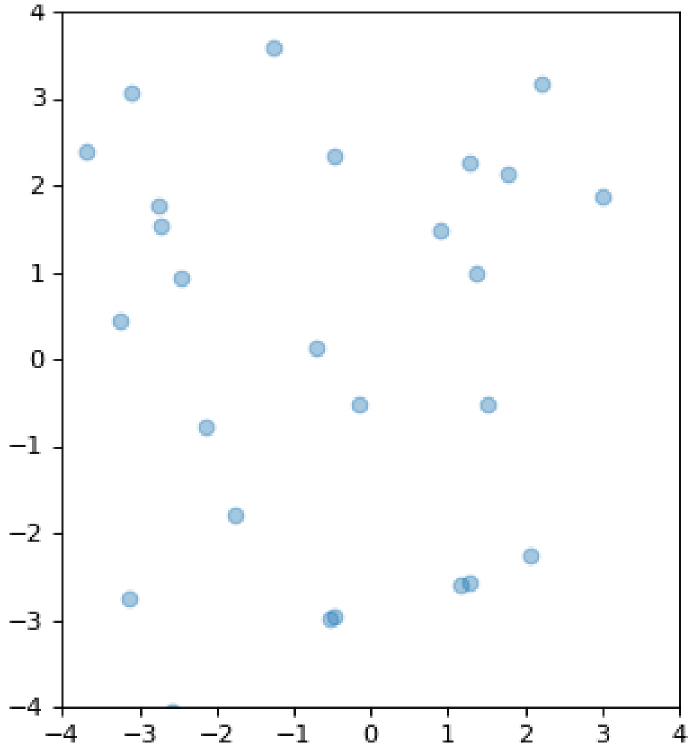
Diffusion OT

Task 0

Before starting the assignment, copy and paste your implementation of `` `2d_plot_ddpm_todo/network.py` `` and `` `image_ddpm_todo/network.py` `` from Assignment 2.

Task 1: Flow Matching with Swiss-Roll

As done in Assignment 1, 2, and 6, you will first implement FM and test it in a simple Swiss-Roll 2D distribution.



Task 1: Flow Matching with Swiss-Roll

Complete the functions `compute_psi_t()` and `step()` in `fm.py`.

```python
def compute_psi_t(self, x1, t, x):
    """
    Compute the conditional flow psi_t(x | x_1).

    Note that time flows in the opposite direction compared to DDPM/DDIM.
    As t moves from 0 to 1, the probability paths shift from a prior distribution p_0(x)
    to a more complex data distribution p_1(x).

    Input:
        x1 (`torch.Tensor`): Data sample from the data distribution.
        t (`torch.Tensor`): Timestep in [0,1).
        x (`torch.Tensor`): The input to the conditional psi_t(x).
    Output:
        psi_t (`torch.Tensor`): The conditional flow at t.
    """
    t = expand_t(t, x1)

    ######## TODO ########
    # DO NOT change the code outside this part.
    # compute psi_t(x)

    psi_t = x1
    ######################

    return psi_t
```

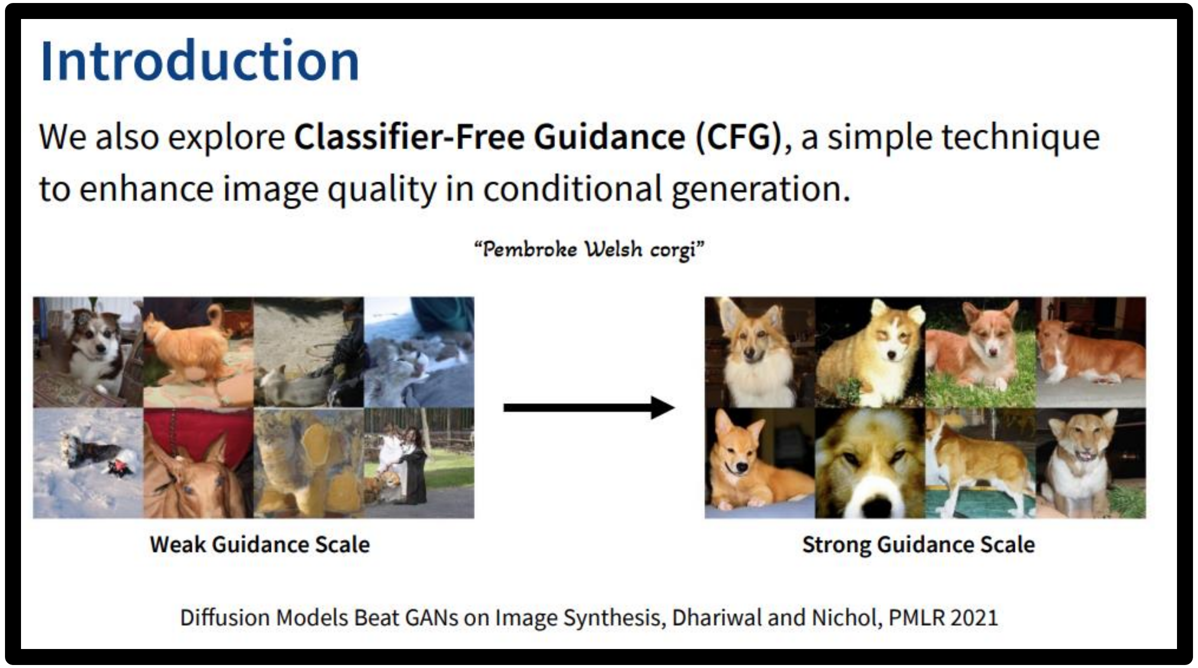
```python
def step(self, xt, vt, dt):
    """
    The simplest ode solver as the first-order Euler method:
    x_next = xt + dt * vt
    """

    ######## TODO ########
    # DO NOT change the code outside this part.
    # implement each step of the first-order Euler method.
    x_next = xt
    ######################

    return x_next
```

Task 2: Image Generation with FM

As done in Assignments 2 and 6, we will sample images using FM with a CFG setup.



Task 2: Image Generation with FM

Finish implementing `sample()` and `get_loss()` functions to work with a CFG setup.

```python
def get_loss(self, x1, class_label=None, x0=None):
    """
    The conditional flow matching objective, corresponding Eq. 23 in the FM paper.
    """
    batch_size = x1.shape[0]
    t = self.fm_scheduler.uniform_sample_t(batch_size).to(x1)
    if x0 is None:
        x0 = torch.randn_like(x1)

    ######## TODO ########
    # DO NOT change the code outside this part.
    # Implement the CFM objective.
    if class_label is not None:
        model_out = self.network(x1, t, class_label=class_label)
    else:
        model_out = self.network(x1, t)

    loss = x1.mean()
    ######################

    return loss
```

What to Submit

Include the following items into a PDF file: {NAME}_{SID}.pdf.

Task 1

- Loss curvature screenshot
- Chamfer distance results of FM sampling with 50 inference steps.
- Visualization of FM sampling.

Task 2

- FID score result obtained with the CFG scale of 7.5.
- At least 8 images generated by Flow Matching.

What to Submit

Create a single ZIP file {NAME}_{SID}.zip including:

- The PDF file, formatted according to the guideline;
- Your implemented code.

Your score will be deducted by 10% for *each* missing item.

Please check carefully!

Grading

You will receive up to 20 points from this assignment.

For Task 1, you will receive:

- 10 points: Achieve CD lower than 40.
- 5 points: Achieve CD between 40 and 60.
- 0 points: Otherwise.

Grading

You will receive up to 20 points from this assignment.

For Task 2, you will receive:

- 10 points: Achieve FID between 30 with CFG=7.5.
- 5 points: Achieve FID between 30 and 50 with CFG=7.5.
- 0 points: Otherwise.

Thank You