HW9

March 31, 2025

Please complete the `NotImplemented` parts of the code cells and write your answers in the markdown cells designated for your response to any questions asked. The tag `# AUTOGRADED` (all caps, with a space after `#`) should be at the beginning of each autograded code cell, so make sure that you do not change that. You are also not allowed to import any new package other than the ones already imported. Doing so will prevent the autograder from grading your code.

For the code submission, run the last cell in the notebook to create the submission zip file. If you are working in Colab, make sure to download and then upload a copy of the completed notebook itself to its working directory to be included in the zip file. Finally, submit the zip file to Gradescope.

After you finish the assignment and fill in your code and response where needed (all cells should have been run), save the notebook as a PDF using the `jupyter nbconvert --to pdf HW9.ipynb` command (via a notebook code cell or the command line directly) and submit the PDF to Gradescope under the PDF submission item. If you cannot get this to work locally, you can upload the notebook to Google Colab and create the PDF there. You can find the notebook containing the instruction for this on Canvas.

If you are running the notebook locally, make sure you have created a virtual environment (using `conda` for example) and have the proper packages installed. We are working with `python=3.10` and `torch>=2`.

Files to be included in submission:

- `HW9.ipynb`
- `model_config.yaml`
- `train_config.yaml`

1 Implement and Train a Diffusion Model on MNIST

[1]:

```python
from typing import Sequence, Tuple, Union
from tqdm import tqdm
import numpy as np
import matplotlib.pyplot as plt
# plt.rcParams.update({'figure.autolayout': True})
from IPython.display import display, clear_output

import torch
from torch import nn, optim
from torch.nn import functional as F
from torch.utils.data import Dataset, DataLoader
from torchvision import datasets
from torchvision.transforms import v2

from google.colab import drive
drive.mount('/content/drive')
import os
os.chdir("/content/drive/MyDrive/24789HW9")

if torch.cuda.is_available():
    Device = 'cuda'
elif torch.backends.mps.is_available():
    Device = 'mps'
else:
    Device = 'cpu'
print(f'Device is {Device}')
```

```
Mounted at /content/drive
Device is cuda
```

1.1 Forward and Reverse Diffusion (25)

Your first task is to implement the forward and reverse diffusion process.

1.1.1 Noise schedule

First we define a tensor containing β_t, α_t, and $\bar{\alpha}_t$. Since the tensors will be used on batched data, you have to make their shapes broadcastable to a batch of data. Here is their relationship:

$$\alpha_t = 1 - \beta_t$$

$$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

In the paper, the diffusion step range is $t = \{1, 2, ..., T\}$. However, since indexing starts from `0` in python, `t` will range from `0` to `T-1` in the code. We use `torch.cumprod` to obtain $\bar{\alpha}$ from α.

1.1.2 Forward diffusion (q)

You should implement the following for a batch of data $\mathbf{x}_0$ and their corresponding diffusion steps t:

$$\mathbf{x}_t = (\sqrt{\bar{\alpha}_t})\mathbf{x}_0 + (\sqrt{1 - \bar{\alpha}_t})\epsilon_q$$

$$\epsilon_q \sim N(\mathbf{0}, \mathbf{1})$$

which samples from this distribution:

$$q(\mathbf{x}_t | \mathbf{x}_0) = N(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{1})$$

Since each sample may have a different t, you should be careful with the shapes and make sure that broadcasting is done correctly. Also remember that ϵ_q should be of the same shape and device as the data.

1.1.3 Reverse diffusion (p)

You should implement the following for a batch of corrupted data $\mathbf{x}_t$ and the diffusion step t:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sqrt{\beta_t}\epsilon_p$$

$$\epsilon_p = \begin{cases} 0 & t = 1(0 \text{ in python}) \\ \sim N(\mathbf{0}, \mathbf{1}) & t > 1 \end{cases}$$

which samples from this distribution (except the final denoising step with $t = 1$ which is deterministic):

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = N\Big(\frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)), \beta_t\mathbf{1}\Big)$$

Here, $\epsilon_\theta(\mathbf{x}_t, t)$ is the output of the neural network that is trained to estimate the noise from the forward diffusion (ϵ_q). Again, remember that indexing starts from `0` in python.

[3]:
```
class Diffusion(nn.Module):

    def __init__(
        self,
        T: int = 1000, # total number of diffusion steps,
        b_0: float = 1e-4,
        b_T: float = 2e-2,
        n_data_dims: int = 3, # number of data dimensions. For example, colored␣
 ↪image data has 3 (channel, height, width)
        ):
        super().__init__()
        self.T = T

        # calculate the 1D tensor for beta containing the values for each␣
 ↪diffusion step
        # using quadratic schedule
        beta = torch.linspace(b_0**0.5, b_T**0.5, T)**2

        # based on n_data_dims, make the shape of beta broadcastable to batched␣
 ↪data
        beta = beta.view(T, *([1] * n_data_dims))

        # calculate alpha and alpha_bar from beta
        # both alpha and alpha_bar have T elements as well, one for each␣
 ↪diffusion step
        alpha = 1. - beta
        alpha_bar = alpha.cumprod(dim=0)

        # register the tensors as buffers to be saved with the model
        # and to be moved to the right device when calling .to(device)
        # You can access them like a normal attribute, like self.alpha
        self.register_buffer('alpha', alpha)
        self.register_buffer('alpha_bar', alpha_bar)
        self.register_buffer('beta', beta)

    @torch.no_grad()
    def forward(
            self,
            x_0: torch.FloatTensor, # (batch_size, *data_shape),
            t: torch.LongTensor, # (batch_size,),
            )-> Tuple[torch.FloatTensor, torch.FloatTensor]: # noisy data and the epsilon used to corrupt it
        """
        for each data sample in the batch, draw a sample from q(x_t|x_0, t)
        according to the schedule and the corresponding diffusion step of each data sample.

        You can index alpha, alpha_bar, or beta with the tensor t directly,
        and get a batch of alpha, alpha_bar, or beta values.

        Returns:
        x_t: torch.FloatTensor, the corrupted batch
        eps_q: torch.FloatTensor, the noise used to corrupt the data
        """
        alpha_bar_t = self.alpha_bar[t]

        # mean of q(x_t|x_0, t)
        mu = torch.sqrt(alpha_bar_t) * x_0

        # std of q(x_t|x_0, t)
        std = torch.sqrt(1 - alpha_bar_t)

        # sample from q using the reparameterization trick
        eps_q = torch.randn_like(x_0)
        x_t = mu + std * eps_q

        return x_t, eps_q

    @torch.inference_mode()
    def reverse(
            self,
            x_t: torch.FloatTensor, # (batch_size, *data_shape),
            t: int,
            eps_theta: torch.FloatTensor, # (batch_size, *data_shape),
            ):
        """
        for a batch of corrupted data x_t and using the estimated noise␣
↪eps_theta,
        sample from p(x_{t-1}|x_t, t)

        Here, t is the same for all samples in the batch.

        Returns:
        x_t_1: torch.FloatTensor, a single-step denoised batch of data
        """
        alpha_t = self.alpha[t]
        alpha_bar_t = self.alpha_bar[t]
        beta_t = self.beta[t]

        # mean of p(x_{t-1}|x_t, t)
        mu = (1. / torch.sqrt(alpha_t)) * (x_t - (beta_t / torch.sqrt(1 -␣
↪alpha_bar_t)) * eps_theta)

        # std of p(x_{t-1}|x_t, t)
        std = torch.sqrt(beta_t)

        # sample from p using the reparameterization trick
        # NOTE: no noise is added at the final denoising step (t=0 -> eps_p=0)
        if t > 0:
            eps_p = torch.randn_like(x_t)
        else:
            eps_p = torch.zeros_like(x_t)
        x_t_1 = mu + std * eps_p

        return x_t_1
```

1.2 Model Architecture (50)

The neural network of a diffusion model is trained to estimate the noise that was used in the forward diffusion. It takes the noisy data (which has been corrupted with noise according to the noise schedule) and the diffusion step explicitly. Your model should be a U-Net (review week 7 for a refresher on U-Net). The diffusion step t should be embedded, and incorporated at every stage of the U-Net by a distinct feedforward network. Follow the instructions given in the comments. You can see the architecture of the original U-Net below. Our U-Net has a similar architecture, with the addition of the embedder and encoders for the diffusion step. The shape of the data is also different, and no cropping is done in the skip connections. We will use average pooling for the downsampling, and an arbitrary choice of activation and hidden channels.

[4]:
```
class ConvBlock(nn.Module):
    """
    a module in the contracting path of the U-Net.
    simply consists of a series of conv, batchnorm, and activation layers.
    """
    def __init__(
            self,
            in_channels: int,
            out_channels: int,
            activation_name: str = 'ReLU',
            n_layers: int = 2,
            batchnorm: bool = False,
            ):
        super().__init__()
        activation = nn.__getattribute__(activation_name)
        self.layers = nn.Sequential()
        for i in range(n_layers):
            self.layers.append(
                nn.Conv2d(
                    in_channels = in_channels if i == 0 else out_channels,
                    out_channels = out_channels,
                    kernel_size = 3,
                    stride = 1,
                    padding = 'same',
                ))
            if batchnorm:
                self.layers.append(nn.BatchNorm2d(out_channels))
            self.layers.append(activation())

    def forward(
            self,
            x: torch.FloatTensor,
            )-> torch.FloatTensor:

        return self.layers(x)


class UpBlock(nn.Module):
    """
    a module in the expanding path of the U-Net.
    First, the input to the block is upconved and concatenated with the skip␣
 ↪connection.
    Then, a series of conv, batchnorm, and activation layers are applied.
    """
    def __init__(
            self,
            in_channels: int,
            skip_channels: int,
            out_channels: int,
            activation_name: str = 'ReLU',
            n_layers: int = 2,
            batchnorm: bool = False,
            ):
        super().__init__()

        self.up = nn.ConvTranspose2d(
            in_channels = in_channels,
            out_channels = out_channels,
            kernel_size = 2,
            stride = 2,
            padding = 0,
            )

        self.layers = ConvBlock(
            in_channels = out_channels + skip_channels,
            out_channels = out_channels,
            activation_name = activation_name,
            n_layers = n_layers,
            batchnorm = batchnorm,
            )

    def forward(
            self,
            x: torch.FloatTensor,
            skip: torch.FloatTensor,
            )-> torch.FloatTensor:

        # upconv x
        x = self.up(x)

        # concatenate the output with the skip connection
        x = torch.cat([skip, x], dim=1)

        # pass x through the main block and return the result
        return self.layers(x)


class FeedForward(nn.Module):
    """
    A simple feedforward neural network to decode the embedded diffusion step␣
 ↪in each stage.
    """
    def __init__(
            self,
            in_features: int,
            out_features: int,
            hidden_sizes: Sequence[int],
            activation_name: str = 'ReLU',
            ):
        super().__init__()
        activation = nn.__getattribute__(activation_name)
        n_layers = len(hidden_sizes)
        self.layers = nn.Sequential()
        for i in range(n_layers):
            self.layers.append(
                nn.Linear(
                    in_features = in_features if i == 0 else hidden_sizes[i-1],
                    out_features = hidden_sizes[i],
                    ))
            self.layers.append(activation())

        self.layers.append(
            nn.Linear(
                in_features = hidden_sizes[-1],
                out_features = out_features,
                ))

    def forward(
            self,
            x: torch.FloatTensor, # (batch_size, in_features)
            )-> torch.FloatTensor: # (batch_size, out_features, 1, 1)
        """
        The output is going to be added to data of shape (B, C, H, W)
        where C = out_features,
        So it has to be broadcastable to the same shape.
        """
        return self.layers(x)[..., None, None]


class Model(nn.Module):

    """
    A U-Net with a corresponding diffusion step decoder for each block.
    """
    def __init__(
            self,
            data_shape: Sequence[int] = [1, 32, 32],

            # diffusion parameters
            T: int = 1000,
            b_0: float = 1e-4,
            b_T: float = 2e-2,

            # diffusion step embedding
            t_embed_dim: int = 128,

            # U-Net architecture
            channels: Sequence[int] = [16, 32, 64, 128],
            n_block_layers: int = 2,
            activation_name: str = 'ReLU',
            batchnorm: bool = False,
            ):
        super().__init__()

        # to be used for data generation
        self.data_shape = data_shape

        # ======================= Diffusion ==========================

        self.diffusion = Diffusion(
            T = T,
            b_0 = b_0,
            b_T = b_T,
            n_data_dims = len(data_shape),
            )

        # =========================== Model ===========================

        n_layers = len(channels)
        self.n_layers = n_layers

        self.blocks = nn.ModuleDict()

        self.t_emebdder = nn.Embedding(
            num_embeddings = T,
            embedding_dim = t_embed_dim,
            )

        self.t_decoder = nn.ModuleDict()
        # An encoder should be assigned to the output of each block in
        # the contracting path or the expanding path.
        # Each encoder takes as input the embedded diffusion step.

        # contracting path
        for i in range(n_layers):
            # example for 4 layers:
            # down_0, down_1, down_2, down_3
            self.blocks[f'down_{i}'] = ConvBlock(
                in_channels = data_shape[0] if i == 0 else channels[i-1],
                out_channels = channels[i],
                activation_name = activation_name,
                n_layers = n_block_layers,
                batchnorm = batchnorm,
            )

            # the output of the t_encoder will be added to the output of the␣
↪block
            self.t_decoder[f'down_{i}'] = FeedForward(
                in_features = t_embed_dim,
                out_features = channels[i],
                hidden_sizes = [channels[i]],
                activation_name = activation_name,
            )

        # expanding path (reverse depth)
        for i in range(n_layers-2, -1, -1):
            # example for 4 layers:
            # up_2, up_1, up_0

            self.blocks[f'up_{i}'] = UpBlock(
                in_channels = channels[i+1],
                skip_channels = channels[i],
                out_channels = channels[i],
                activation_name = activation_name,
                n_layers = n_block_layers,
                batchnorm = batchnorm,
            )

            # the output of the t_encoder will be added to the output of the␣
↪block
            self.t_decoder[f'up_{i}'] = FeedForward(
                in_features = t_embed_dim,
                out_features = channels[i],
                hidden_sizes = [channels[i]],
                activation_name = activation_name,
            )

        # final output layer to get the estimated noise tensor
        # the output should have the same shape as the input data
        # use kernel_size = 1, stride = 1, padding = 0
        self.out = nn.Conv2d(
            in_channels = channels[0],
            out_channels = data_shape[0],
            kernel_size = 1,
            stride = 1,
            padding = 0,
        )
```

```
  def forward(
          self,
          x: torch.FloatTensor, # (batch_size, *data_shape)
          t: Union[int, torch.LongTensor], # (batch_size,)
          ) -> torch.FloatTensor: # (batch_size, *data_shape)
      """
      Inputs:
          x: a batch of corrputed data
          t: the corresponding diffusion step for each sample in the batch

      returns:
          eps_theta: the estimated noise tensor used to corrupt the data in␣
↪the forward diffusion.
      """

      if isinstance(t, int):
          # create a LongTensor of shape (batch_size,) from t, on the same␣
↪device as x_0
          t = torch.tensor(len(x)*[t], device=x.device, dtype=torch.long)

      # embed the diffusion step
      t_embedded = self.t_emebdder(t)

      # to store the skip connections
      skips = []

      # contracting path
      for i in range(self.n_layers):

          # pass data through the block
          x = self.blocks[f'down_{i}'](x)

          # append the result to skips, to be used in the expanding path
          # except for the last down block (deepest layer)
          if i < self.n_layers - 1:
            skips.append(x)

          # encode t_embedded with the corresponding decoder
          # and add it channel-wise to data
          x = x + self.t_decoder[f'down_{i}'](t_embedded)

          # downsample data with F.avg_pool2d and kernel_size=2
          # except for the last down block (deepest layer)
          if i < self.n_layers - 1:
            x = F.avg_pool2d(x, kernel_size=2)
```

```
        # expanding path (reverse depth)
        for i in range(self.n_layers-2, -1, -1):

            # pass the data and the corresponding skip connection to the block
            x = self.blocks[f'up_{i}'](x, skips[i])

            # encode t_embedded with the corresponding decoder
            # and add it channel-wise to the data
            x = x + self.t_decoder[f'up_{i}'](t_embedded)

        # pass data through the final convolutional layer
        eps_theta = self.out(x)

        return eps_theta


    @torch.inference_mode()
    def generate(
        self,
        n_samples: int,
        device: str = Device,
        )-> torch.FloatTensor: # (n_samples, *data_shape):
        """
        Sample a noise tensor of the right shape and device.
        Execute the reverse diffusion process using the model.

        Returns:
           xs of shape (T, n_samples, *data_shape)
           the full batch of denoised samples at each diffusion step
           starting from pure noise and ending with the final generated samples.
        """
        self.eval().to(device)

        # start from pure noise of the right shape and device
        x = torch.randn((n_samples, *self.data_shape), device=device)

        # for the sake of visualization, we will store the data throughout␣
 ↪reverse diffusion
        xs = [x]

        # denoise step-by-step by sampling from p(x_{t-1}|x_t, t)
        # t = T-1, T-2, ..., 1, 0
        for t in range(self.diffusion.T-1, -1, -1):

            # get the model output
            eps_theta = self.forward(x, t)
            # do one step of reverse diffusion
            x = self.diffusion.reverse(x, t, eps_theta)

            # append the denoised data to xs
            xs.append(x)

        return torch.stack(xs)
```

1.3 Tracking and visualization

[5]:
```
class Diffusion_Tracker:
    """
    Logs and plots different loss terms of a GAN during training.
    """
    def __init__(
            self,
            n_iters: int,
            plot_freq: Union[int, None] = None, # plot every plot_freq␣
 ↪iterations
            ):

        self.losses = []

        self.plot = plot_freq is not None
        self.iter = 0
        self.n_iters = n_iters

        if self.plot:
            self.plot_freq = plot_freq
            self.plot_results()


    def plot_results(self):
        self.fig, self.ax = plt.subplots(figsize=(12, 4))

        # Score plot:
        self.loss_curve, = self.ax.plot(
            range(1, self.iter+1),
            self.losses,
            )

        self.ax.set_xlim(0, self.n_iters+1)
        self.ax.set_ylim(0, 1)
        self.ax.set_xlabel('Iteration')
        self.ax.set_ylabel('Loss')
        self.ax.set_title('Diffusion Learning Curve')
        self.ax.grid(linestyle='--')

        self.samples_fig, self.samples_axes = plt.subplots(5, 8, figsize=(8,␣
↪5), sharex=True, sharey=True)
        self.sample_axes = self.samples_axes.flat
        self.samples = []
        for ax in self.sample_axes:
            ax.axis('off')
            self.samples.append(ax.imshow(np.zeros((32, 32)), cmap='gray',␣
↪vmin=0, vmax=1))


    def update(
            self,
            loss: float,
            ):
        self.losses.append(loss)
        self.iter += 1

        if self.plot and self.iter % self.plot_freq == 0:

            # score plot:
            self.loss_curve.set_data(range(1, self.iter+1), self.losses)
            self.ax.relim()
            self.ax.autoscale_view()

            self.samples_fig.suptitle(f'Generated Samples at Iteration {self.
↪iter}')

            self.fig.canvas.draw()
            clear_output(wait=True)
            display(self.fig)
            display(self.samples_fig)


    def get_samples(
            self,
            samples: torch.FloatTensor, # (n_samples, *output_shape)
            ):
        for sample, sample_img in zip(samples, self.samples):
            sample_img.set_data(sample.clip(0, 1).detach().squeeze().cpu().
↪numpy())
```

1.4 Training (10)

[6]:
```
@torch.enable_grad()
def train(
        model: Model,
        train_dataset: Dataset,
        device = Device,
        plot_freq: int = 100,

        optimizer_name: str = 'Adam',
        optimizer_config: dict = dict(),
        lr_scheduler_name: Union[str, None] = None,
        lr_scheduler_config: dict = dict(),

        n_iters: int = 5000,
        batch_size: int = 32,
        ):

    model.train().to(device)

    tracker = Diffusion_Tracker(
        n_iters = n_iters,
        plot_freq = plot_freq,
        )

    train_loader = DataLoader(train_dataset, batch_size=batch_size,␣
 ↪shuffle=True)
    optimizer: optim.Optimizer = optim.__getattribute__(optimizer_name)(model.
 ↪parameters(), **optimizer_config)
    if lr_scheduler_name is not None:
        lr_scheduler: optim.lr_scheduler._LRScheduler = optim.lr_scheduler.
 ↪__getattribute__(lr_scheduler_name)(optimizer, **lr_scheduler_config)

    iter = 0
    iter_pbar = tqdm(range(n_iters), desc='Iters', unit='iter', leave=True)

    while iter < n_iters:

        for x, _ in train_loader:

            model.train()

            x = x.to(device)

            # sample a random diffusion step for each sample in the batch
            t = torch.randint(0, model.diffusion.T, (x.shape[0],),␣
 ↪device=device)
```

```
            # forward diffusion
            x_t, eps_q = model.diffusion.forward(x, t)

            # model's forward pass
            eps_theta = model(x_t, t)

            # calculate the loss
            loss = F.mse_loss(eps_theta, eps_q)

            loss.backward()
            optimizer.step()
            optimizer.zero_grad()

            if lr_scheduler_name == 'ReduceLROnPlateau':
                lr_scheduler.step(loss.item())
            elif lr_scheduler_name is not None:
                lr_scheduler.step()

            # ==================== Logging ====================
            iter += 1
            iter_pbar.update(1)
            iter_pbar.set_postfix_str(f'loss: {loss.item():.6f}')
            if iter % plot_freq == 0:
                gen_samples = model.generate(40)[-1]
                tracker.get_samples(gen_samples)
            tracker.update(loss.item())
            if iter >= n_iters:
                break
```

1.5 Dataset

[7]:

```
train_dataset = datasets.MNIST(
    root = 'MNIST',
    train = True,
    download = True,
    # transform the data to torch.Tensor and scale it to [0, 1]
    transform = v2.Compose([
        # convert to tensor and scale to [0, 1]
        v2.ToImage(),
        v2.Pad(2), # to make the images 32x32
        v2.ToDtype(torch.float32, scale=True),
    ])
    )
```

```
100%|      | 9.91M/9.91M [00:00<00:00, 10.5MB/s]
100%|      | 28.9k/28.9k [00:00<00:00, 361kB/s]
```

```
100%|          | 1.65M/1.65M [00:00<00:00, 3.18MB/s]
100%|          | 4.54k/4.54k [00:00<00:00, 8.71MB/s]
```

1.6 Find and train a good model (15)

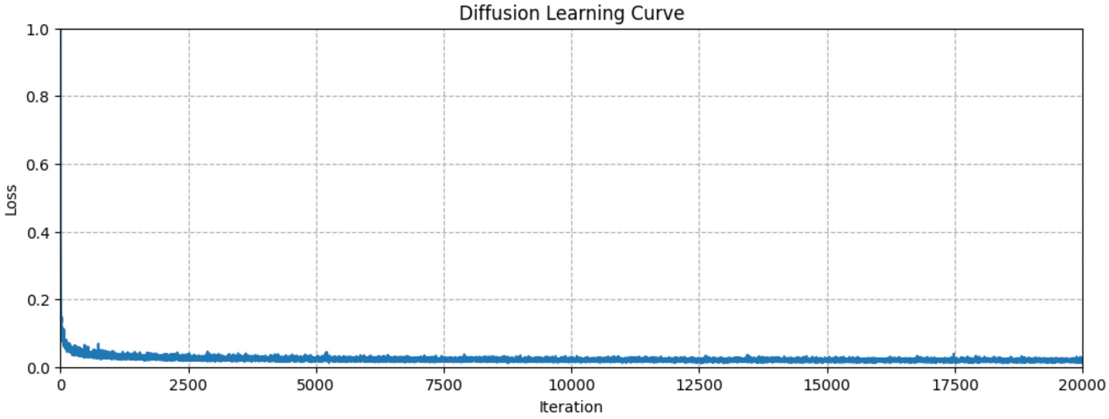
Your grade will depend on the quality and diversity of your generated samples. If almost all your samples are nice, you will get full points. Diffusion models take some time to train. While your loss may not seem to improve, your sample quality might improve with more training, so be parient.

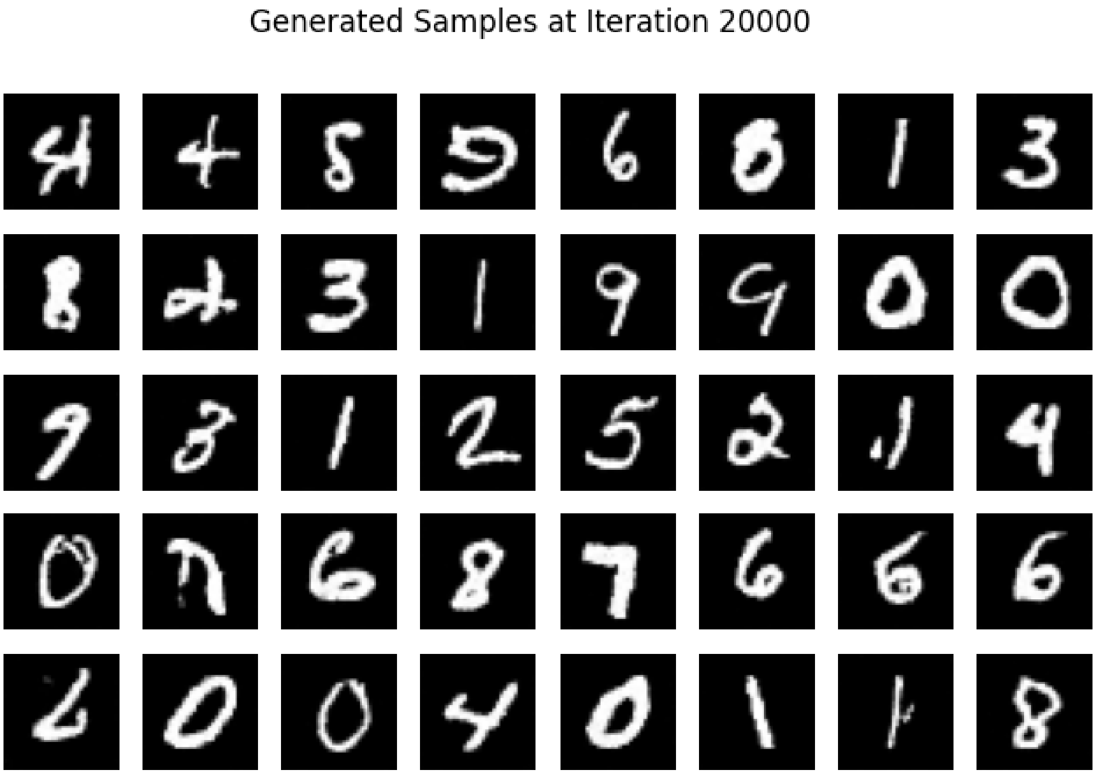
[16]:

```python
# YOUR CODE

model_config = dict(
    # diffusion parameters (we suggest to not change these)
    T = 1000,
    b_0 = 1e-4,
    b_T = 2e-2,

    # model parameters
    t_embed_dim = 256,
    channels = [32, 64, 128, 256],
    n_block_layers = 3,
    activation_name = 'ReLU',
    batchnorm = True,
    )

train_config = dict(
    optimizer_name = 'Adam',
    optimizer_config = {'lr': 1e-3},
    lr_scheduler_name = 'StepLR',
    lr_scheduler_config = {'step_size': 1000, 'gamma': 0.95},

    n_iters = 20000,
    batch_size = 64,
    )
```

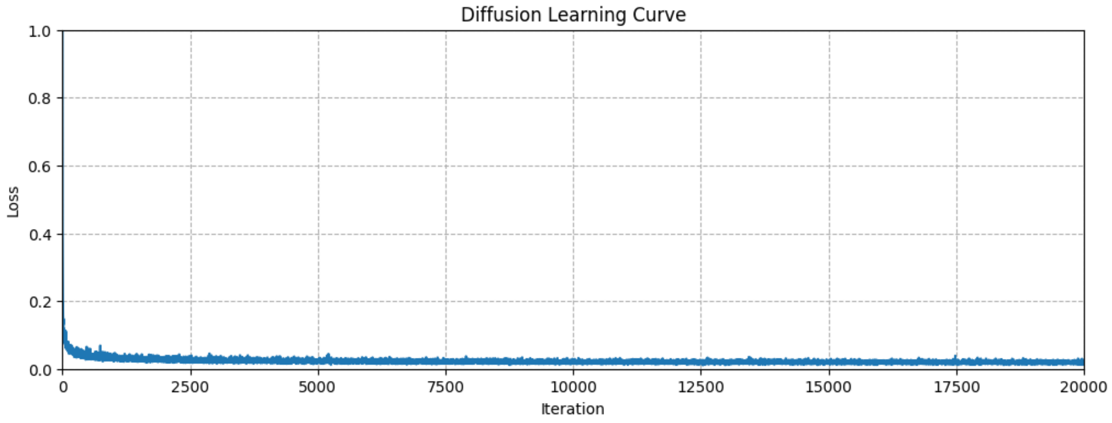
[17]:

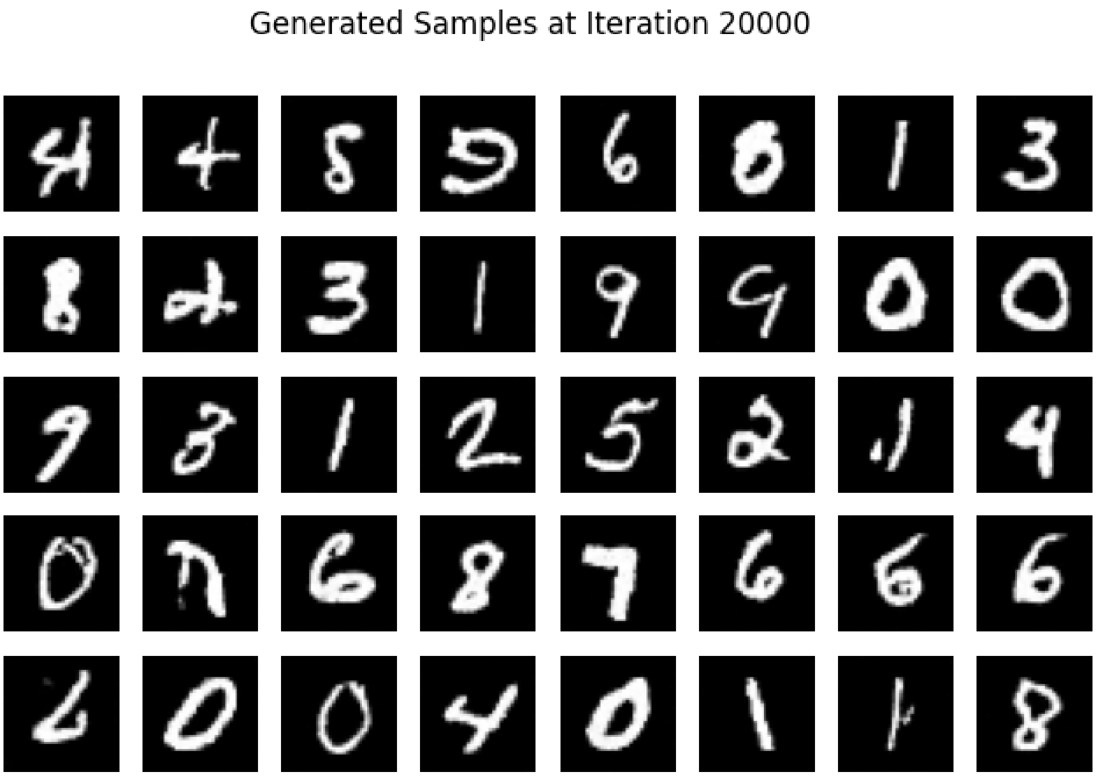
```python
if __name__ == '__main__':
    model = Model(
        data_shape = [1, 32, 32],
        **model_config,
        )
    train(
        model = model,
        train_dataset = train_dataset,
        device = Device,
        plot_freq = 500,
        **train_config,
        )
```





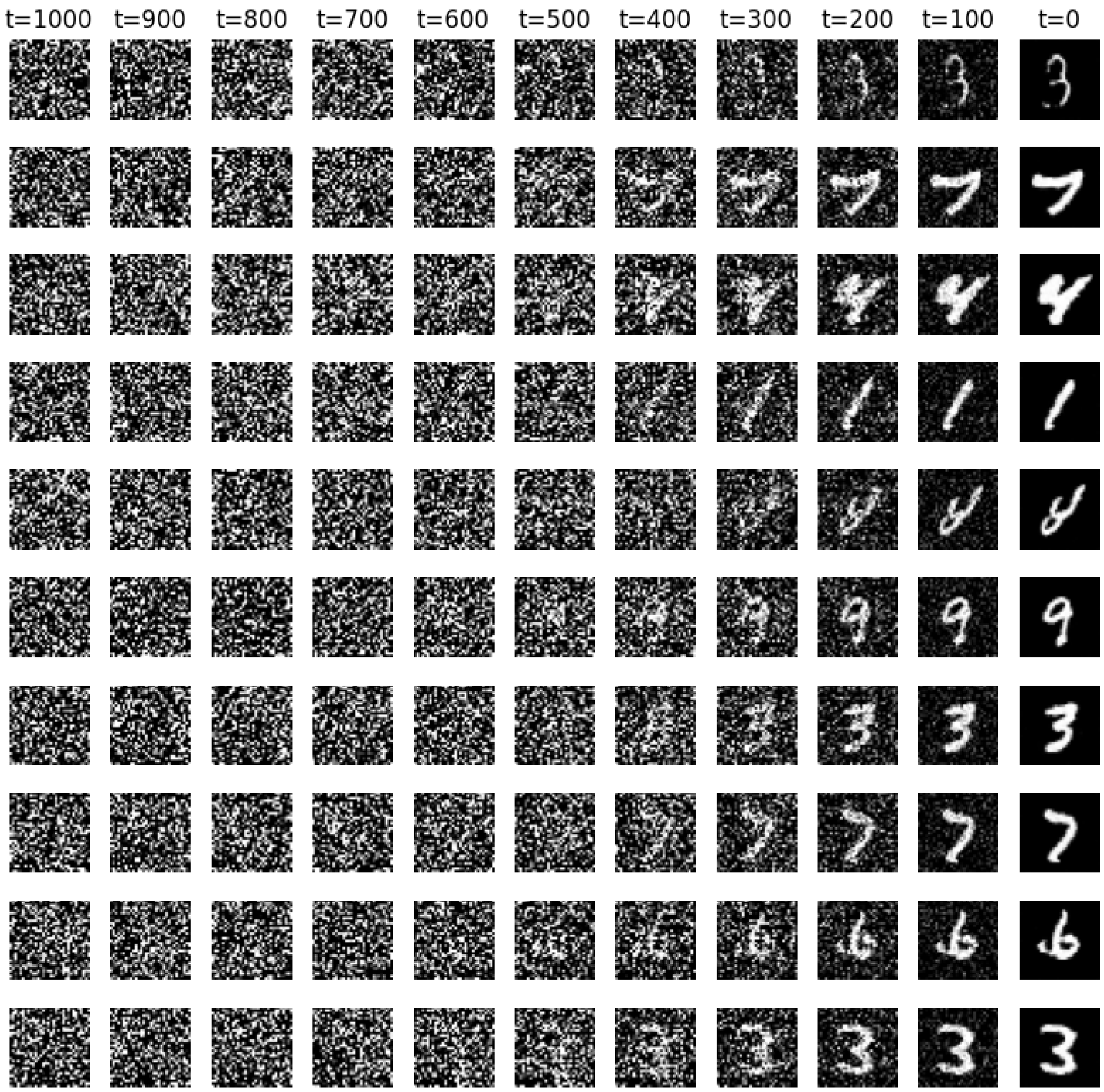
```
Iters: 100%|      | 20000/20000 [24:45<00:00, 13.46iter/s, loss: 0.022789]
```





1.7 Visualize reverse diffusion

[18]:
```
gen_samples = model.generate(10)

def plot_reverse_diffusion(
        xs: torch.FloatTensor, # (T+1, n_samples, *data_shape)
        n_steps: int = 10,
        ):
    n_samples = xs.shape[1]
    ts = list(range(0, model.diffusion.T+1, model.diffusion.T//n_steps))
    fig, axes = plt.subplots(n_samples, n_steps+1, figsize=(8, 8), sharex=True,␣
 ↪sharey=True)
    for sample_idx, ax in enumerate(axes):
        for t_idx, a in enumerate(ax):
            a.axis('off')
            a.imshow(xs[ts[t_idx], sample_idx].squeeze().detach().cpu().
 ↪numpy(), cmap='gray', vmin=0, vmax=1)
            if sample_idx == 0:
                a.set_title(f't={model.diffusion.T-ts[t_idx]}')
    plt.tight_layout()
    plt.show()

plot_reverse_diffusion(gen_samples, n_steps=10)
```



1.8 Zip files for submission

[19]:

```
import os
import yaml
import zipfile

def save_yaml(config: dict, path: str):
    with open(path, 'w') as f:
        yaml.dump(config, f, sort_keys=False, default_flow_style=None)


def zip_files(output_filename: str, file_paths: list):
    with zipfile.ZipFile(output_filename, 'w') as zipf:
        for file_path in file_paths:
            zipf.write(file_path, os.path.basename(file_path))


save_yaml(model_config, 'model_config.yaml')
save_yaml(train_config, 'train_config.yaml')

submission_files = ['HW9.ipynb', 'model_config.yaml', 'train_config.yaml']
zip_files('HW9_submission.zip', submission_files)
```

```
[ ]: !apt-get install texlive texlive-xetex texlive-latex-extra pandoc
```

```
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc-data
poppler-data
  preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-webrick ruby-
xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive-base texlive-
binaries
  texlive-fonts-recommended texlive-latex-base texlive-latex-recommended
texlive-pictures
  texlive-plain-generic tipa xfonts-encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java texlive-
luatex
  pandoc-citeproc context wkhtmltopdf librsvg2-bin groff ghc nodejs php python
libjs-mathjax
  libjs-katex citation-style-language-styles poppler-utils ghostscript fonts-
japanese-mincho
  | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-
arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv | postscript-
viewer perl-tk xpdf
  | pdf-viewer xzdec texlive-fonts-recommended-doc texlive-latex-base-doc
python3-pygments
```

```
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl texlive-latex-
extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex texlive-
pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc
pandoc-data
  poppler-data preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-
webrick ruby-xmlrpc
  ruby3.0 rubygems-integration t1utils teckit tex-common tex-gyre texlive
texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base texlive-latex-
extra
  texlive-latex-recommended texlive-pictures texlive-plain-generic texlive-xetex
tipa
  xfonts-encodings xfonts-utils
0 upgraded, 59 newly installed, 0 to remove and 29 not upgraded.
Need to get 202 MB of archives.
After this operation, 728 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
[2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all
0.4.11-1 [2,171 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17
[33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all
20200910-1 [6,367 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common
all 9.55.0~dfsg1-0ubuntu5.11 [753 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64
1.38-4ubuntu1 [60.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64
0.35-15build2 [16.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64
0.19-3build2 [64.7 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64
9.55.0~dfsg1-0ubuntu5.11 [5,031 kB]
```

```
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6
amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64
1.0.2-1build4 [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64
2.13.1-1 [1,221 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-lmodern all
2.004.5-6.1 [4,532 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-noto-mono all
20201225-1build1 [397 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-texgyre all
20180621-3.1 [10.2 MB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libapache-pom-java
all 18-1 [4,720 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-
gfm0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [115 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcmark-gfm-
extensions0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [25.1 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-parent-
java all 43-1 [10.8 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-logging-
java all 1.2-2 [60.3 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfontenc1 amd64
1:1.1.4-1build3 [14.7 kB]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libptexenc1
amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration
all 1.18 [5,336 B]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64
3.0.2-7ubuntu2.8 [50.1 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-rubygems all
3.3.5-2 [228 kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby amd64 1:3.0~exp1
[5,100 B]
Get:28 http://archive.ubuntu.com/ubuntu jammy/main amd64 rake all 13.0.6-2 [61.7
kB]
Get:29 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-webrick
all 1.7.0-3ubuntu0.1 [52.1 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-xmlrpc all
0.3.2-1ubuntu0.1 [24.9 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libruby3.0
amd64 3.0.2-7ubuntu2.8 [5,113 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsynctex2
amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:34 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libteckit0 amd64
2.5.11+ds1-1 [421 kB]
```

```
Get:35 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexlua53
amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexluajit2
amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libzzip-0-13 amd64
0.13.72+dfsg.1-1.1 [27.0 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all
2.004.5-6.1 [9,471 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc-data all
2.9.2.1-3ubuntu2 [81.8 kB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc amd64
2.9.2.1-3ubuntu2 [20.3 MB]
Get:43 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style
all 12.2-1ubuntu1 [185 kB]
Get:44 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64
1.41-4build2 [61.3 kB]
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64
2.5.11+ds1-1 [699 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all
20180621-3.1 [6,209 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-
binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive all
2021.20220204-1 [14.3 kB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:55 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:56 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:57 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:58 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
```

```
Get:59 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 202 MB in 5s (40.3 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 126210 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
…
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2.1_all.deb …
Unpacking fonts-lato (2.0-2.1) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.11-1_all.deb …
Unpacking poppler-data (0.4.11-1) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.17_all.deb …
Unpacking tex-common (6.17) …
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack …/04-fonts-urw-base35_20200910-1_all.deb …
Unpacking fonts-urw-base35 (20200910-1) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/05-libgs9-common_9.55.0~dfsg1-0ubuntu5.11_all.deb …
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.11) …
Selecting previously unselected package libidn12:amd64.
Preparing to unpack …/06-libidn12_1.38-4ubuntu1_amd64.deb …
Unpacking libidn12:amd64 (1.38-4ubuntu1) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/07-libijs-0.35_0.35-15build2_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-15build2) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/08-libjbig2dec0_0.19-3build2_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.19-3build2) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/09-libgs9_9.55.0~dfsg1-0ubuntu5.11_amd64.deb …
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.11) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack …/11-libwoff1_1.0.2-1build4_amd64.deb …
Unpacking libwoff1:amd64 (1.0.2-1build4) …
Selecting previously unselected package dvisvgm.
Preparing to unpack …/12-dvisvgm_2.13.1-1_amd64.deb …
Unpacking dvisvgm (2.13.1-1) …
Selecting previously unselected package fonts-lmodern.
```

```
Preparing to unpack …/13-fonts-lmodern_2.004.5-6.1_all.deb …
Unpacking fonts-lmodern (2.004.5-6.1) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/14-fonts-noto-mono_20201225-1build1_all.deb …
Unpacking fonts-noto-mono (20201225-1build1) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/15-fonts-texgyre_20180621-3.1_all.deb …
Unpacking fonts-texgyre (20180621-3.1) …
Selecting previously unselected package libapache-pom-java.
Preparing to unpack …/16-libapache-pom-java_18-1_all.deb …
Unpacking libapache-pom-java (18-1) …
Selecting previously unselected package libcmark-gfm0.29.0.gfm.3:amd64.
Preparing to unpack …/17-libcmark-gfm0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcmark-gfm-
extensions0.29.0.gfm.3:amd64.
Preparing to unpack …/18-libcmark-gfm-
extensions0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb …
Unpacking libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack …/19-libcommons-parent-java_43-1_all.deb …
Unpacking libcommons-parent-java (43-1) …
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack …/20-libcommons-logging-java_1.2-2_all.deb …
Unpacking libcommons-logging-java (1.2-2) …
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack …/21-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/22-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/23-rubygems-integration_1.18_all.deb …
Unpacking rubygems-integration (1.18) …
Selecting previously unselected package ruby3.0.
Preparing to unpack …/24-ruby3.0_3.0.2-7ubuntu2.8_amd64.deb …
Unpacking ruby3.0 (3.0.2-7ubuntu2.8) …
Selecting previously unselected package ruby-rubygems.
Preparing to unpack …/25-ruby-rubygems_3.3.5-2_all.deb …
Unpacking ruby-rubygems (3.3.5-2) …
Selecting previously unselected package ruby.
Preparing to unpack …/26-ruby_1%3a3.0~exp1_amd64.deb …
Unpacking ruby (1:3.0~exp1) …
Selecting previously unselected package rake.
Preparing to unpack …/27-rake_13.0.6-2_all.deb …
Unpacking rake (13.0.6-2) …
Selecting previously unselected package ruby-net-telnet.
```

```
Preparing to unpack …/28-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-webrick.
Preparing to unpack …/29-ruby-webrick_1.7.0-3ubuntu0.1_all.deb …
Unpacking ruby-webrick (1.7.0-3ubuntu0.1) …
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack …/30-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb …
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack …/31-libruby3.0_3.0.2-7ubuntu2.8_amd64.deb …
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.8) …
Selecting previously unselected package libsynctex2:amd64.
Preparing to unpack …/32-libsynctex2_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack …/33-libteckit0_2.5.11+ds1-1_amd64.deb …
Unpacking libteckit0:amd64 (2.5.11+ds1-1) …
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack …/34-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/35-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/36-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/37-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/38-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package lmodern.
Preparing to unpack …/39-lmodern_2.004.5-6.1_all.deb …
Unpacking lmodern (2.004.5-6.1) …
Selecting previously unselected package pandoc-data.
Preparing to unpack …/40-pandoc-data_2.9.2.1-3ubuntu2_all.deb …
Unpacking pandoc-data (2.9.2.1-3ubuntu2) …
Selecting previously unselected package pandoc.
Preparing to unpack …/41-pandoc_2.9.2.1-3ubuntu2_amd64.deb …
Unpacking pandoc (2.9.2.1-3ubuntu2) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/42-preview-latex-style_12.2-1ubuntu1_all.deb …
Unpacking preview-latex-style (12.2-1ubuntu1) …
Selecting previously unselected package t1utils.
```

```
Preparing to unpack …/43-t1utils_1.41-4build2_amd64.deb …
Unpacking t1utils (1.41-4build2) …
Selecting previously unselected package teckit.
Preparing to unpack …/44-teckit_2.5.11+ds1-1_amd64.deb …
Unpacking teckit (2.5.11+ds1-1) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/45-tex-gyre_20180621-3.1_all.deb …
Unpacking tex-gyre (20180621-3.1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/46-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/47-texlive-base_2021.20220204-1_all.deb …
Unpacking texlive-base (2021.20220204-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/48-texlive-fonts-recommended_2021.20220204-1_all.deb …
Unpacking texlive-fonts-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/49-texlive-latex-base_2021.20220204-1_all.deb …
Unpacking texlive-latex-base (2021.20220204-1) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/50-texlive-latex-recommended_2021.20220204-1_all.deb …
Unpacking texlive-latex-recommended (2021.20220204-1) …
Selecting previously unselected package texlive.
Preparing to unpack …/51-texlive_2021.20220204-1_all.deb …
Unpacking texlive (2021.20220204-1) …
Selecting previously unselected package libfontbox-java.
Preparing to unpack …/52-libfontbox-java_1%3a1.8.16-2_all.deb …
Unpacking libfontbox-java (1:1.8.16-2) …
Selecting previously unselected package libpdfbox-java.
Preparing to unpack …/53-libpdfbox-java_1%3a1.8.16-2_all.deb …
Unpacking libpdfbox-java (1:1.8.16-2) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/54-texlive-pictures_2021.20220204-1_all.deb …
Unpacking texlive-pictures (2021.20220204-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/55-texlive-latex-extra_2021.20220204-1_all.deb …
Unpacking texlive-latex-extra (2021.20220204-1) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/56-texlive-plain-generic_2021.20220204-1_all.deb …
Unpacking texlive-plain-generic (2021.20220204-1) …
Selecting previously unselected package tipa.
Preparing to unpack …/57-tipa_2%3a1.3-21_all.deb …
Unpacking tipa (2:1.3-21) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/58-texlive-xetex_2021.20220204-1_all.deb …
Unpacking texlive-xetex (2021.20220204-1) …
```

```
Setting up fonts-lato (2.0-2.1) …
Setting up fonts-noto-mono (20201225-1build1) …
Setting up libwoff1:amd64 (1.0.2-1build4) …
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libijs-0.35:amd64 (0.35-15build2) …
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libfontbox-java (1:1.8.16-2) …
Setting up rubygems-integration (1.18) …
Setting up libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Setting up fonts-urw-base35 (20200910-1) …
Setting up poppler-data (0.4.11-1) …
Setting up tex-common (6.17) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) …
Setting up libjbig2dec0:amd64 (0.19-3build2) …
Setting up libteckit0:amd64 (2.5.11+ds1-1) …
Setting up libapache-pom-java (18-1) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) …
Setting up t1utils (1.41-4build2) …
Setting up libidn12:amd64 (1.38-4ubuntu1) …
Setting up fonts-texgyre (20180621-3.1) …
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up ruby-webrick (1.7.0-3ubuntu0.1) …
Setting up libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Setting up fonts-lmodern (2.004.5-6.1) …
Setting up libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Setting up pandoc-data (2.9.2.1-3ubuntu2) …
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.11) …
Setting up teckit (2.5.11+ds1-1) …
Setting up libpdfbox-java (1:1.8.16-2) …
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.11) …
Setting up preview-latex-style (12.2-1ubuntu1) …
Setting up libcommons-parent-java (43-1) …
Setting up dvisvgm (2.13.1-1) …
Setting up libcommons-logging-java (1.2-2) …
Setting up xfonts-utils (1:7.7+6build2) …
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up pandoc (2.9.2.1-3ubuntu2) …
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) …
Setting up texlive-base (2021.20220204-1) …
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
Setting up tex-gyre (20180621-3.1) …
Setting up texlive-plain-generic (2021.20220204-1) …
Setting up texlive-latex-base (2021.20220204-1) …
Setting up texlive-latex-recommended (2021.20220204-1) …
Setting up texlive-pictures (2021.20220204-1) …
Setting up texlive-fonts-recommended (2021.20220204-1) …
Setting up tipa (2:1.3-21) …
Setting up texlive (2021.20220204-1) …
Setting up texlive-latex-extra (2021.20220204-1) …
Setting up texlive-xetex (2021.20220204-1) …
Setting up rake (13.0.6-2) …
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.8) …
Setting up ruby3.0 (3.0.2-7ubuntu2.8) …
Setting up ruby (1:3.0~exp1) …
Setting up ruby-rubygems (3.3.5-2) …
Processing triggers for man-db (2.10.2-1) …
Processing triggers for mailcap (3.70+nmu1ubuntu1) …
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) …
Processing triggers for libc-bin (2.35-0ubuntu3.8) …
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libumf.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a
symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link

Processing triggers for tex-common (6.17) …
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
Building format(s) --all.
        This may take some time…
```