CS 236, Fall 2021
Midterm Exam

This exam is worth 90 points. You have 3.5 hours to complete and submit it. You are allowed to consult notes, books, the internet, and use a laptop. But no communication is allowed. Good luck!

Stanford University Honor Code

The Honor Code is the University's statement on academic integrity written by students in 1921. It articulates University expectations of students and faculty in establishing and maintaining the highest standards in academic work:

- The Honor Code is an undertaking of the students, individually and collectively:
  - that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
  - that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.
- The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
- While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Question	**Score**	**Question**	**Score**
1	/ 0	5	/ 10
2	/ 18	6	/ 10
3	/ 10	7	/ 12
4	/ 10	8	/ 20

Total score: / 90

Note: Partial credit will be given for partially correct answers. Zero points will be given to answers left blank.

1. **[0 points total] Stanford Honor Code**

   - This exam is open-notes. This means you can reference notes, lectures slides, and other resources. If you use resources outside of notes and lecture slides, please cite your source.
   - No form of collaboration is allowed.
   - Please do not openly discuss anything about the contents of the exam (e.g. on Ed, Slack, in-person, etc.) with other people, both students and non-students, during the exam period and until exam grades have been released. This includes asking questions and receiving real-time assistance from Q&A answer sites such as Stack Overflow, Chegg, Yahoo Answers, etc.
   - By taking this exam, you attest to following the Stanford Honor Code: I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honor Code.

2. **[18 points total] True/False**

For each of the statements below, state True or False. Explain your answer for full points.

(a) [**2 points**] Without any independence assumptions, the number of parameters for a tabular autoregressive model depends on the exact choice of variable ordering.

(b) [**2 points**] Consider a discrete autoregressive model over greyscale images with pixel intensity values in $\{0, 1, \ldots, 255\}$. Using at most 256 forward passes, it is always possible to exactly compute the conditional distribution of *any* single missing pixel given the values for all the other pixels.

(c) [**2 points**] Given a latent variable model $p_\theta(x, z)$ and a fixed choice of observation $\bar{x}$, you can interpret the function $p_\theta(\bar{x}, z)$ as an unnormalized distribution with respect to z, and whose partition function is $p_\theta(\bar{x})$.

(d) [**2 points**] Because the variational autoencoder objective contains a reconstruction term, an optimally-trained VAE will always make use of the latent space and thus have non-zero Kl-divergence to the prior: $D_{\mathrm{KL}}(q(z \mid x) \parallel p(z)) > 0$.

[Note: in this question, we define an optimally-trained VAE to be one with a tight ELBO (i.e., equality holds between the ELBO and the log-likelihood) and whose marginal distribution matches the data distribution.]

(e) [**2 points**] Any continuous autoregressive flow model $p_\theta(\mathbf{x}_{1:n}) = \prod_{i=1}^{n} p_\theta(\mathbf{x}_i \mid \mathbf{x}_{<i})$, where each $p_\theta(\mathbf{x}_i \mid \mathbf{x}_{<i})$ is a conditional probability density function, can be represented as a flow model with a uniform prior.

(f) [**2 points**] When training a GAN model with Minimax Loss, the gradient with respect to the generator parameters will be zero if we fix the discriminator so that it outputs a constant value for all inputs.

(g) [**2 points**] Let R be a rotation matrix (i.e., such that $R^T R = I$) and $X = f(Z)$ be a flow model where $Z \sim \mathcal{N}(0, I)$ is distributed as a Gaussian with unit covariance I. Then $g(Z) = f(RZ)$ is another flow model that achieves the same likelihood as f on any dataset.

(h) [**2 points**] A normalizing flow model will map an observed random variable X to a lower dimensional latent variable Z.

(i) [**2 points**] Training an EBM always requires estimating its partition function.

3. **[10 points total] Change of Variables**

(a) [**5 points**] You are dealing with a 32×32 greyscale image dataset whose pixel intensities are *real-valued* in the interval $[0, 255]$. A common pre-processing procedure is to scale your data by $1/127.5$ and then shifting it by -1, so that your data lies in the interval $[-1, 1]$, before training your Gaussian autoregressive model $p_\theta(\mathbf{x})$, where $\mathbf{x}$ has dimensionality 32×32. You do so and report a test set log-likelihood of

$$\frac{1}{N}\sum_{i=1}^{N} \ln p_\theta(\mathbf{x}^{(i)}) = 32.5, \tag{1}$$

where $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$ is your test set and each $\mathbf{x}^{(i)}$ is a processed $[-1, 1]^{32\times32}$ image. However, Reviewer 2 requests that you report your model's test set log-likelihood in the original $[0, 255]^{32\times32}$ space for your report to be comparable with the literature. What is your test set log-likelihood in the original $[0, 255]^{32\times32}$ space? Report your value to the third significant digit in scientific notation. Explain how you got your answer for full credit.

(b) [**5 points**] Given a univariate Normal (i.e., Gaussian) random variable Y, its exponentiation $X = \exp(Y)$ is said to have a Log-Normal distribution. If Y is distributed according to $\mathcal{N}(\mu, \sigma^2)$, then we denote $X = \exp(Y)$ as being distributed according to $\text{LN}(\mu, \sigma^2)$. Using the definition of the Gaussian probability density function for p_Y and the change-of-variables formula that relates p_Y to p_X, prove that the probability density function for $X \sim \text{LN}(\mu, \sigma^2)$ is

$$p_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right). \tag{2}$$

4. **[10 points total] KL Divergence and Bounds**

   In the VAE lectures, we have seen how addition/subtraction of a KL-divergence term can yield a bound (e.g. subtracting a KL term from the log-likelihood in a latent variable model yields the Evidence Lower Bound). We shall apply this same technique of adding/subtracting a KL term to answer the following questions.

   (a) **[5 points]** Given a joint distribution $p(x, z)$, show that

   $$\mathbb{E}_{p(x,z)}\left[\ln \frac{p(x,z)}{p(x)p(z)}\right] \le \mathbb{E}_{p(z)} D_{\mathrm{KL}}(p(x \mid z) \parallel q(x)) \tag{3}$$

   for any choice of q. Explicitly show the KL-divergence term you are adding/subtracting in your work.

   (b) **[5 points]** Given a joint distribution $p(x, z)$, show that

   $$\mathbb{E}_{p(x,z)}\left[\ln \frac{p(x,z)}{p(x)p(z)}\right] \ge -\mathbb{E}_{p(z)}\left[\ln p(z)\right] + \mathbb{E}_{p(z,x)}[\ln q(z \mid x)] \tag{4}$$

   for any choice of q. Explicitly show the KL-divergence term you are adding/subtracting in your work.

5. **[10 points total] MCMC-Based Training of Latent Variable Models**

   So far, we have seen how variational methods (i.e. ELBO maximization) can be used to train the latent variable model $p_\theta(x, z)$ where x is observed and z is latent. In this question, we shall consider a popular alternative called Markov Chain Monte Carlo (MCMC). For the purposes of this question, we shall simply treat MCMC as a black-box method that—with enough computation time—reliably allows us to sample from (but not compute!) the posterior $p_\theta(z \mid x)$. Fortunately, the ability to sample from the posterior (even if we cannot compute it) is sufficient for constructing an unbiased estimate of the gradient of the log-likelihood, thanks to the following identity,

$$\nabla_\theta \ln p_\theta(x) = \mathbb{E}_{p_\theta(z|x)} \nabla_\theta \ln p_\theta(x, z). \tag{5}$$

   Prove this identity using the formula for the gradient of the logarithm function (log-derivative trick): $\nabla_\theta \ln p_\theta(x) = \frac{1}{p_\theta(x)} \cdot \nabla_\theta p_\theta(x)$.

6. **[10 points total] Variational Perspective to Energy-Based Models**

   In this question, we will consider energy-based models from a variational perspective.

   (a) **[5 points]** Consider an EBM with an unnormalized distribution $\tilde{p}_\theta(x)$ and partition function $Z(\theta) = \int \tilde{p}_\theta(x) \mathrm{d}x$. Computing the log-partition function $\ln Z(\theta)$ is usually intractable. So we shall look to bounding this quantity instead. In particular, if we introduce a proposal distribution $q(x)$ that is easy to compute and sample from, we can construct the following lower bound for the log-partition function,

   $$\ln Z(\theta) \geq \mathbb{E}_{q(x)} \left[\ln \frac{\tilde{p}_\theta(x)}{q(x)} \right]. \tag{6}$$

   Prove that this bound holds for any choice of q. It may help to notice the strong resemblance between this expression and the Evidence Lower Bound for a latent variable model.

   (b) **[5 points]** Consider again the EBM with an unnormalized $\tilde{p}_\theta(x)$ and partition function $Z(\theta) = \int \tilde{p}_\theta(x) \mathrm{d}x$. Note that, when normalized, $p_\theta(x) = \tilde{p}_\theta(x)/Z(\theta)$. So far, we have learned from class that gradient-based optimization of the EBM's log-likelihood requires computing

   $$\nabla_\theta \ln p_\theta(x) = \nabla_\theta \ln \tilde{p}_\theta(x) - \mathbb{E}_{p_\theta(x)} \nabla_\theta \ln \tilde{p}_\theta(x), \tag{7}$$

   We now take a variational perspective to this expression by introducing the variational family $\mathcal{Q}$, which we shall denote as the set of all possible distributions over x. Prove that

   $$\nabla_\theta \ln p_\theta(x) = \nabla_\theta \ln \tilde{p}_\theta(x) - \mathbb{E}_{q^*(x)} \nabla_\theta \ln \tilde{p}_\theta(x), \tag{8}$$

   where

   $$q^*(x) = \arg\max_{q \in \mathcal{Q}} \mathbb{E}_{q(x)} \left[\ln \frac{\tilde{p}_\theta(x)}{q(x)} \right]. \tag{9}$$

   You may make use of and do not have to prove Equation (**??**). [Hint: Prove that $q^* = p_\theta$.]

7. **[12 points total] Masked Autoencoder**

In this question, we shall design a mask within a Masked Autoencoder. Our MADE model takes as input $\mathbf{x} \in \mathbb{R}^3$ and outputs predictions $\hat{\mathbf{x}}_i$ conditional on all preceding input dimensions $\mathbf{x}_{<i}$. We have provided the masks M_1, M_2, M_4, and M_5 in the figure below.

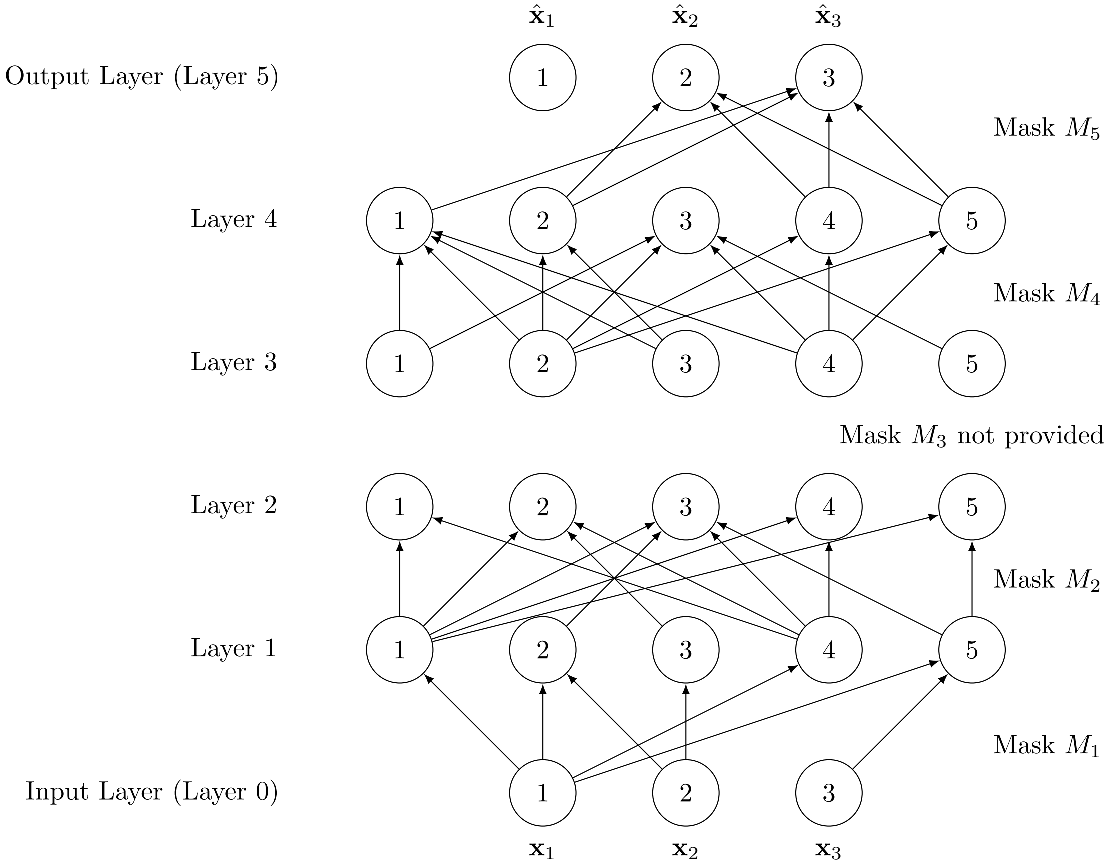

Figure 1: MADE Model with masks provided for M_1, M_2, M_4, and M_5.

The index for each neuron is provided in the figure. Each mask M_ℓ is a binary matrix, where the element $(M_\ell)_{ij} = 1$ if and only if the i^{th} neuron of layer $\ell - 1$ points to the j^{th} neuron of the subsequent layer ℓ, otherwise $(M_\ell)_{ij} = 0$.

We have not provided the mask M_3. Your objective is to design the densest possible binary mask $M_3 \in \{0,1\}^{5\times 5}$ that preserves the autoregressive property, $p(x) = \prod_{i=1}^{3} p(x_i \mid x_{<i})$, in our MADE model. In other words, we want M_3 to be a valid mask (i.e., preserving the autoregressive property) that has as many non-zero elements as possible. For your convenience, we are also providing the matrix multiplications for $M_1 \cdot M_2$ and $M_4 \cdot M_5$, which we denote as matrices A and B,

$$A = M_1 \cdot M_2 = \begin{pmatrix} 2 & 2 & 4 & 2 & 2 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \qquad B = M_4 \cdot M_5 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 4 \\ 0 & 1 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & 0 \end{pmatrix}. \tag{10}$$

Answer the following questions related to the determination of the densest valid mask for M_3.

(a) [**2 points**] What is the index of the largest-indexed Layer 0 neuron that has a path to the 2nd neuron of Layer 2? For full credit, describe how to get your answer using matrix A, and without relying on Figure **??**.

(b) [**2 points**] What is the index of the smallest-indexed Layer 2 neuron that has a path from the 3rd neuron of Layer 0? For full credit, describe how to get your answer using matrix A, and without relying on Figure **??**.

(c) [**8 points**] Based on our specific mask choices for M_1, M_2, M_4, M_5, express the densest valid mask for M_3 as an adjacency list, according to the following example format:

Layer 0 Neuron 1 points to Layer 1 Neurons: $1, 2, 4, 5$
Layer 0 Neuron 2 points to Layer 1 Neurons: $2, 3$
Layer 0 Neuron 3 points to Layer 1 Neurons: 5

For full credit, describe how to get your answer using the matrices A and B, and without relying on Figure **??**.

8. **[20 points total] GAN Loss and Weighted Jensen-Shannon Divergence**

This problem explores how the GAN objective function $\mathcal{L}$ relates to the Jensen-Shannon divergence. Given a distribution p that we wish to model, recall the GAN optimization problem

$$\min_q \max_D \mathcal{L}(q, D) = \min_q \max_D \mathbb{E}_{p(x)} \left[\ln D(x)\right] + \mathbb{E}_{q(x)} \left[\ln \left(1 - D(x)\right)\right], \tag{11}$$

where q is the generative model and D is the discriminator. In class, we showed that if the discriminator is optimized over all possible discriminative functions, the optimal discriminator $D^*(x)$ reduces the GAN objective to

$$\mathcal{L}(q, D^*) = 2 \cdot D_{\text{JS}}(p \parallel q) - \ln(4). \tag{12}$$

where $D_{\text{JS}}(p \parallel q)$ is the JS-divergence.

(a) **[10 points]** For any choice of weight $\pi \in (0, 1)$, we can define a π-weighted version of the Jensen-Shannon divergence as

$$D_{\text{JS}_\pi}(p \parallel q) = \pi \cdot D_{\text{KL}}\Big(p \,\Big\|\, \pi p + (1-\pi) q\Big) + (1-\pi) \cdot D_{\text{KL}}\Big(q \,\Big\|\, \pi p + (1-\pi) q\Big). \tag{13}$$

For notational simplicity, we shall refer to the π-weighted JS-divergence simply as the weighted JS-divergence henceforth. A natural consideration is whether the weighted JS-divergence is an f-divergence. For the following choice of generator function f,

$$f(u) = \pi u \ln u - (\pi u + 1 - \pi) \ln \left(\pi u + 1 - \pi\right), \tag{14}$$

prove that

$$D_{\text{JS}_\pi}(p \parallel q) = D_f(p \parallel q), \tag{15}$$

where $D_f(p \parallel q) = \mathbb{E}_{q(x)}[f(\frac{p(x)}{q(x)})]$ denotes the f-divergence.

(b) **[10 points]** Since the weighted JS-divergence is an f-divergence, we can cast the weighted JS-divergence as a variational divergence problem instead. Recall that the Fenchel conjugate for any function f is

$$f^*(t) = \sup_{u \in \text{dom}(f)} ut - f(u). \tag{16}$$

For the weighted JS-divergence, the Fenchel conjugate for its generator function is

$$f^*(t) = (1-\pi) \ln \left(\frac{1-\pi}{1 - \pi \cdot \exp\left(\frac{t}{\pi}\right)} \right), \tag{17}$$

where the domain of f^* is $t < -\pi \ln \pi$. Based on this and the equations from Question (**??**), prove that

$$D_{\text{JS}_\pi}(p \parallel q) \geq \mathbb{E}_{p(x)} \left[\ln D(x)\right] - \mathbb{E}_{q(x)} \left[(1-\pi) \ln \left(\frac{1-\pi}{1 - \pi D(x)^{\frac{1}{\pi}}} \right) \right], \tag{18}$$

for any choice of function $D : \mathcal{X} \to (0, (\frac{1}{\pi})^\pi)$. This gives us a GAN-like objective to approximately minimize the weighted JS-divergence.