CS 236, Fall 2023
Midterm Exam

This exam is worth 95 points. You have 3 hours to complete and submit it. You are allowed to consult notes, books and use a laptop. But no communication or network access is allowed. Use of Large Language Models (LLMs) is also not 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	/ 20	5	/ 11
2	/ 10	6	/ 10
3	/ 18	7	/ 16
4	/ 10		

Total score: **/ 95**

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

Name: _____ SUNet ID: _____@stanford.edu

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.

Sign: _____

1. **[20 points total] True/False**

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

   (a) **[2 points] Autoregressive models**:
   For discrete autoregressive models, the order of variables in the chain rule factorization cannot affect the minimal number of parameters required to represent exactly the joint distribution.
   Answer: False. Some prediction tasks (corresponding to certain orderings) are easier to solve and require less parameters (for example, when there is conditional independence).

   (b) **[2 points] Autoregressive Model**: Let $p(\mathbf{x})$ be an arbitrary autoregressive model over a set of discrete random variables. The probability assigned by p to a sequence $\mathbf{x} = (x_1, x_2, \dots x_n)$ of length n is always at least as large as the probability assigned by p to any sequence $(x_1, x_2, \dots x_n, x_{n+1})$ of length $n+1$ of which $\mathbf{x}$ is a prefix.
   Answer: True. The probability for the longer sequence is obtained by multiplying by a number smaller or equal than one (a probability).

   (c) **[2 points] VAE**: In a VAE with two binary latent variables $z_1 \in \{0, 1\}$, $z_2 \in \{0, 1\}$ and a Gaussian decoder $p(x|z_1, z_2)$, the marginal likelihood $p(x)$ is intractable to compute.
   Answer: False, it's a mixture of gaussians and can be brute forced.

   (d) **[2 points] ELBO**: Let the inference distribution of a VAE be $q_\phi(z|x)$ and prior distribution be $p(z)$, the gap between the MLE objective and the ELBO lower bound for a datapoint x is exactly $D_{\mathrm{KL}}(q_\phi(z|x)\|p(z))$.
   Answer: False.

$$\begin{aligned}
\log p_\theta(x) &= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x)] \\
&= \mathbb{E}_{q_\phi(z|x)}[\log \frac{p_\theta(x,z)}{p_\theta(z|x)}] \\
&= \mathbb{E}_{q_\phi(z|x)}[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}] + \mathbb{E}_{q_\phi(z|x)}[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}]
\end{aligned}$$

   The gap is $D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z|x))$.

   (e) **[2 points] ELBO**: Alice and Bob train two VAEs on the same dataset. The VAEs are identical, except for the priors over the latents which are different. Alice's model achieves a better (higher) ELBO on the training set. This implies Alice's model also achieves higher log-likelihood on the training set.
   Answer: False. Higher ELBO does not necessarily correlate with higher log-likelihood.

   (f) **[2 points] Normalizing-Flow**: A normalizing flow based model with latent variables $\mathbf{z} \in \mathbb{R}^{100}$ can be used to model (with tractable exact likelihood evaluation) any random vector $\mathbf{x}$ of dimension less than or equal to 100.
   Answer: False. The inputs and outputs of a normalizing flow should have the same shape.

(g) **[2 points] GAN**: A VAE with a Gaussian prior and a Gaussian decoder can be trained as the generator in a GAN (discarding the encoder and using a separate classifier as a discriminator).

Answer: True.

(h) **[2 points] GAN**: A conditional GAN can be implemented by adding the class information as an additional input to the Generator $G(z)$, without changes to the Discriminator $D(x)$.

Answer: False. Should add to both Generator and Discriminator.

(i) **[2 points] EBM**: For general energy-based models, the energy $E(x)$ must be positive for every x.

Answer: False. $E(x)$ can be arbitrary.

(j) **[2 points] EBM**: There exist energy-based models that cannot be equivalently described as an autoregressive model, regardless of the variable ordering and parameterization of the conditionals.

Answer: False, AR factorization is fully general and works for any ordering.

2. **[10 points total] Comparison of Models**

So far we discussed five major types of generative models: autoregressive models (AR), variational autoencoders (VAE), flow models (Flow), generative adversarial networks (GAN), and energy-based models (EBM). In the following questions, we will compare their strengths and weaknesses. Brief justifications (1 sentence) are sufficient for full points.

(a) **[2 points]** Which of these models can be used to model both discrete and continuous data?

Answer: AR, VAE, EBM

(b) **[2 points]** Suppose we are interested in exactly evaluating the likelihood of a data point under the trained model. In which of these models can we exactly evaluate a data point's likelihood in an efficient way (i.e., in a time polynomial in the number of dimensions of a sample, such as in linear time)?

Answer: Autoregressive and flow models allow for exact likelihood evaluation.

(c) **[2 points]** Which of these models can be used to model conditional distributions (e.g., trained to map captions to images given a suitable paired dataset)?

Answer: All of them

(d) **[2 points]** Which of these models can be used to solve "in-painting" tasks (i.e., fill in missing values in a datapoint such as completing a sentence or an image where some parts have been masked out), assuming access to infinite compute?

Answer: all

(e) **[2 points]** Suppose we are interested in learning a latent representation for new data points. Which of these models are most appropriate for this task, and why?

Answer: In a VAE, $q_\phi(z|x)$ can serve as an encoder. In a flow model, $f^{-1}(x)$ can serve as an encoder. GAN can also learn representations via Bi-GAN. EBM can have latents like an RBM or DBM.

3. **[18 points total] Variational Autoencoders Basics** In this question, we will consider a Variational Autoencodeer model where as usual $\mathbf{x}$ denotes observed variables and $\mathbf{z}$ denotes latent variables. For each of the following questions briefly explain your answer for full points.

   (a) **[3 points]** Suppose we are training a VAE where the prior $p_\theta(\mathbf{z})$ is parameterized by a Generative Adversarial Network instead of standard Gaussian (i.e., samples from the prior are produced by a trainable generator G_θ), and where the encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$ and decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ are Gaussian. Can we train this VAE by optimizing the ELBO?

   Answer: No, we need tractable likelihood for the prior. In other words, to evaluate the ELBO objective we need to be able to evaluate $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z})p_\theta(\mathbf{x} \mid \mathbf{z})$, but evaluating the $p_\theta(\mathbf{z})$ term is intractable for a GAN.

   (b) **[3 points]** Suppose we have trained a VAE parameterized by ϕ for encoder and θ for decoder and obtained a global optimum of the ELBO objective (with infinite data from p_{data}, perfect optimization, arbitrary flexible decoder and encoder). Note this implies that the KL divergence between $p_{\text{data}}(\mathbf{x})$ and $p_\theta(\mathbf{x})$ is zero. Suppose we start from a real datapoint $\overline{\mathbf{x}}$ sampled from p_{data}, sample $\mathbf{z}' \sim q_\phi(\mathbf{z} \mid \overline{\mathbf{x}})$ by feeding $\overline{\mathbf{x}}$ to the encoder, then draw a sample $\mathbf{x}' \sim p_\theta(\mathbf{x}|\mathbf{z}')$ by feeding $\mathbf{z}'$ to the decoder. Is $\mathbf{x}'$ distributed according to p_{data}?

   Answer: Yes. Because the ELBO is tight, $KL(q_\phi(z \mid x)||p_\theta(z \mid x)) = 0$ is zero, and $KL(p_{data}(x)||p_\theta(x)) = 0$ is also zero. In other words, $KL(p_{data}(x)q_\phi(z \mid x)||p_\theta(x)p_\theta(z \mid x)) = 0 = KL(p_{data}(x)q_\phi(z \mid x)||p_\theta(z)p_\theta(x \mid z))$. This implies that z' is marginally distributed as $p_\theta(z)$, and $\mathbf{x}'$ is marginally distributed as p_{data}.

   (c) **[3 points]** Alice and Bob are training two identical VAEs on the same data distribution. They both claim to have obtained a global optimum of the ELBO objective. When they evaluate their models on a test datapoint $\mathbf{x}$, they find their respective encoders map $\mathbf{x}$ to different distributions over the latents. Is this possible if they truly have both obtained a global optimum of the ELBO objective?

   Answer: Yes, it's not identifiable, i.e. there could be multiple equally good solutions

   (d) **[3 points]** Let $\mathbf{x} \in \mathbb{R}^D$ denote the inputs, $\mathbf{z}$ the latent variables, $p_\theta(\mathbf{x}|\mathbf{z})$ the generative model, $p(\mathbf{z})$ the prior and $q_\phi(\mathbf{z}|\mathbf{x})$ as the basic inference model representing a Gaussian distribution $\mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$. Consider an alternative inference model $q_{\phi,A}(\mathbf{z}|\mathbf{x})$ also representing a Gaussian distribution with parameters $\mathcal{N}(\mathbf{z}; \mu_\phi(A\mathbf{x}), \text{diag}(\sigma_\phi(A\mathbf{x})^2))$ where A is a (trainable) $D \times D$ matrix.

   Show that the best evidence lower bound we can achieve with $q_{\phi,A}$ is at least as tight as the best one we can achieve with q_ϕ, i.e.,

   $$\max_{\theta,\phi,A} \text{ELBO}(\mathbf{x}; p_\theta, q_{\phi,A}) \geq \max_{\theta,\phi} \text{ELBO}(\mathbf{x}; p_\theta, q_\phi)$$

   where

   $$\text{ELBO}(\mathbf{x}; p, q) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\mathbf{x})]$$

   and A is optimized over the set of all $D \times D$ matrices.

   Answer: Choose A to be the identity matrix

(e) **[3 points]** Let $\mathbf{x} \in \mathbb{R}^D$ denote the inputs, $\mathbf{z}$ the latent variables, $p_\theta(\mathbf{x}|\mathbf{z})$ the generative model, $p(\mathbf{z})$ the prior and $q_\phi(\mathbf{z}|\mathbf{x})$ as the basic inference model representing a Gaussian distribution $\mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$. Consider an alternative inference model $q_{\phi,B}(\mathbf{z}|\mathbf{x})$:

- Sample $\tilde{\mathbf{z}} \sim \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$
- Compute $\mathbf{z} = B\tilde{\mathbf{z}}$

where B is a trainable matrix. Can you efficiently optimize the objective $\text{ELBO}(\mathbf{x}; p_\theta, q_{\phi,B})$ corresponding to this alternative inference model?

Answer: Yes, can still use the reparameterization trick.$\mathbf{z}$ is Gaussian $\mathcal{N}(\mathbf{z}; B\mu_\phi(\mathbf{x}), B\text{diag}(\sigma_\phi(\mathbf{x})^2)B^T)$

(f) **[3 points]** Compare $\max_{\theta,\phi,A} \text{ELBO}(\mathbf{x}; p_\theta, q_{\phi,A})$ to $\max_{\theta,\phi,B} \text{ELBO}(\mathbf{x}; p_\theta, q_{\phi,B})$, is one always larger than the other without assumptions on μ_ϕ and σ_ϕ?

Answer: Not comparable in general. The A in $q_{\phi,A}$ cannot be taken outside of the network without assumptions.

4. **[10 points total] Normalizing Flow Models:**

   (a) **[2 points]** Let $Z \sim \text{Uniform}[-2, 3]$ (a uniform random variable over the interval $[-2, 3]$) and $X = Z^3$. What is $p_X(1)$?.
   Answer: By change of variables $p_X(x) = \frac{1}{3}x^{-\frac{2}{3}} p_Z(\sqrt[3]{x}) = 1/15$

   (b) **[2 points]** Let $Z \sim \text{Uniform}[-2, 3]$ and $X = Z^2$. Is this a valid normalizing flow model?
   Answer: No, the transformation is not invertible

   (c) **[2 points]** Let $Z \sim \text{Uniform}[0, 3]$ and $X = Z^2$. Is this a valid normalizing flow model?
   Answer: Yes, there is a unique inverse

   (d) **[4 points] Change of Variable with Polar Coordinates.** Let X and Y be random variables representing Cartesian coordinates in a plane. Let R and Θ be random variables representing the corresponding radius R and angle Θ in a polar coordinate system. Then, Cartesian coordinates X and Y and polar coordinates are related by $X = R\cos(\Theta)$ and $Y = R\sin(\Theta)$, *i.e.*, $(X, Y) = f(R, \Theta) = (R\cos(\Theta), R\sin(\Theta))$. Assume X and Y have joint probability density $p_{X,Y}(x, y)$. Using the change-of-variables formula, derive the joint probability density function $p_{R,\Theta}(r, \theta)$ in terms of $p_{X,Y}$.
   Hint: Potentially useful formula:

$$\frac{\partial}{\partial\theta}\cos(\theta) = -\sin\theta, \quad \frac{\partial}{\partial\theta}\sin(\theta) = \cos\theta$$

   Answer: Jacobian:

$$|J| = \frac{\partial f^{-1}(x, y)}{\partial(r, \theta)} = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{vmatrix} = r$$

$$p_{R,\Theta}(r, \theta) = |J| \cdot p_{X,Y}(x, y) = r \cdot p_{X,Y}(r\cos\theta, r\sin\theta).$$

5. **[11 points total] Variational Recurrent Neural Network:** Here we explore a recurrent version of the VAE for the purpose of modelling sequences. Drawing inspiration from simpler dynamic Bayesian networks (DBNs) such as HMMs and Kalman filters, this variational recurrent neural network (VRNN) augments a traditional RNN model with a sequence of latent variables, similar to a VAE. VRNN additionally models the dependendencies between these latent random variables over time with another RNN.

   A VRNN model defines a joint distribution $p(x_1, \cdots, x_T, z_1, \cdots, z_T)$ over two groups of random variables $x_1, \cdots, x_T$ and $z_1, \cdots, z_T$. You can think of $x_1, \cdots, x_T$ as the elements of a time series, and $z_1, \cdots, z_T$ as latent variables that are also indexed by time. For brevity, we denote $\mathbf{x}_{\leq t} = (x_1, \cdots, x_t)$, $\mathbf{x}_{<t} = (x_1, \cdots, x_{t-1})$, and we similarly define $\mathbf{z}_{\leq t}$ and $\mathbf{z}_{<t}$. In a VRNN, the joint is factorized as

$$p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p_{\theta_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p_{\theta_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$$

   where $p_{\theta_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t})$ and $p_{\theta_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$ are parameterized as RNNs, and $\theta = (\theta_1, \theta_2)$. As an edge case, $p_{\theta_1}(x_1|\mathbf{z}_{\leq 1}, \mathbf{x}_{<1}) = p_{\theta_1}(x_1|z_1)$ and $p_{\theta_2}(z_1|\mathbf{x}_{<1}, \mathbf{z}_{<1}) = p_{\theta_2}(z_1)$.

   (a) **[2 points]** How many parameters are needed to specify the model as a function of T?
   Answer: Cardinality of θ, constant with respect to T.

   (b) **[2 points]** In what order are the variables sampled to generate data from the model?
   Answer: $z_1, x_1, z_2, x_2, \cdots, z_T, x_T$

   (c) **[2 points]** Consider an alternative model

$$p_\psi(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p_{\psi_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p_{\psi_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{>t})$$

   where $p_{\psi_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t})$ and $p_{\psi_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{>t})$ are parameterized as RNNs. Is this a valid autoregressive model?
   Answer: No, there are cyclical dependencies (no chain rule)

   (d) **[4 points] Learning:** Because the marginal likelihood $p_\theta(\mathbf{x}_{\leq T})$ is intractable to compute, we need to resort to an ELBO for training. We consider the following encoder:

$$q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T}) = \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$$

   where $q_\phi(z_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ is parameterized using another RNN.
   Show that the following ELBO for the VRNN described above

$$E_{\mathbf{z}_{\leq T} \sim q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})} \left[\log \left(\frac{p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T})}{q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})} \right) \right]$$

is equivalent to

$$E_{\mathbf{z}_{\leq T}\sim q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\sum_{t=1}^{T}(-D_{KL}\left(q_\phi(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})||p_{\theta_2}(z_t|\mathbf{x}_{<t},\mathbf{z}_{<t})\right)+\log\ p_{\theta_1}(x_t|\mathbf{z}_{\leq t},\mathbf{x}_{<t}))\right]$$

where $D_{KL}(p||q)$ denotes KL divergence.

Answer:

$$ELBO = E_{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\log\left(\frac{p(\mathbf{x}_{\leq T},\mathbf{z}_{\leq T})}{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\right)\right]$$

$$\log\left(\frac{p(\mathbf{x}_{\leq T},\mathbf{z}_{\leq T})}{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\right)=\sum_{t=1}^{T}(\log\ p(x_t|\mathbf{z}_{\leq t},\mathbf{x}_{<t})))+\sum_{t=1}^{T}\left[\log\left(\frac{p(z_t|\mathbf{x}_{<t},\mathbf{z}_{<t})}{q(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})}\right)\right]$$

Notice, that since q can be factorised as described in the question:

$$E_{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\log\left(\frac{p(z_t|\mathbf{x}_{<t},\mathbf{z}_{<t})}{q(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})}\right)\right]=E_{q(\mathbf{z}_{<t}|\mathbf{x}_{<t})}\left[E_{q(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})}\left[\log\left(\frac{p(z_t|\mathbf{x}_{<t},\mathbf{z}_{<t})}{q(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})}\right)\right]\right]$$

This is because q can be factorised for different timesteps and variables for timesteps $> t$ don't affect the expectation (can be marginalised out). Then we can first marginalise out t (the inner expectation) and then marginalise out the timesteps $i < t$.

$$E_{q(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})}\left[\log\left(\frac{p(z_t|\mathbf{x}_{<t},\mathbf{z}_{<t})}{q(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})}\right)\right]=-D_{KL}(q(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})||p(z_t|\mathbf{x}_{<t},\mathbf{z}_{<t}))$$

Combining both the terms, we get

$$ELBO = E_{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\sum_{t=1}^{T}(-D_{KL}(q(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})||p(z_t|\mathbf{x}_{<t},\mathbf{z}_{<t}))+\log p(x_t|\mathbf{z}_{\leq t},\mathbf{x}_{<t}))\right]$$

(e) **[1 points]** Assume the conditionals $q_\phi(z_t|\mathbf{x}_{\leq t},\mathbf{z}_{<t})$ in the encoder RNN are Gaussians (with mean and variance computed by the RNN). Can the ELBO be optimized using the reparameterization trick?

Answer: Yes, the sampling step can be rewritten as a deterministic transformation of standard Gaussians by unrolling the RNN.

6. **[10 points total] GAN: Saturating and Non-Saturating Loss**

   Generative Adversarial Networks suffer from the saturation problem, which is when the generator G_θ cannot train as fast as the discriminator D_ϕ and stops learning. To overcome the saturation problem, a non-saturating loss is often used. Non-saturating loss is a subtle modification of the original generator loss, where the generator maximizes the probability of generated images being real rather than minimizing the probability of generated images being fake.

   (a) **[2 points] Non-saturating loss:** Recall the original GAN generator loss for a single sample $\mathbf{z}$:

   $$\mathcal{L}_S(\theta) = \log\left(1 - D_\phi\left(G_\theta\left(\mathbf{z}\right)\right)\right).$$

   Next, consider the following non-saturating loss:

   $$\mathcal{L}_{NS}(\theta) = -\log\left(D_\phi\left(G_\theta\left(\mathbf{z}\right)\right)\right).$$

   Show that for fixed $\mathbf{z}$ and D_ϕ:

   $$\arg\min_\theta \mathcal{L}_S(\theta) = \arg\min_\theta \mathcal{L}_{NS}(\theta)$$

   Answer:

   $$\begin{aligned}\arg\min_\theta \log\left(1 - D_\phi\left(G_\theta\left(\mathbf{z}\right)\right)\right) &= \arg\max_\theta D_\phi\left(G_\theta\left(\mathbf{z}\right)\right) \\ &= \arg\max_\theta \log(D_\phi\left(G_\theta\left(\mathbf{z}\right)\right)) = \arg\min_\theta -\log(D_\phi\left(G_\theta\left(\mathbf{z}\right)\right))\end{aligned}$$

   (b) **[2 points] Gradient of the saturating Loss:** What is the derivative of the saturating loss $\mathcal{L}_S(\theta)$ with respect to $D_\phi(G_\theta(\mathbf{z}))$?

   Answer:

   $$\frac{\partial \mathcal{L}_S}{\partial D_\phi\left(G_\theta\left(z\right)\right)} = -\frac{1}{1 - D_\phi\left(G_\theta\left(z\right)\right)}.$$

   (c) **[2 points] Gradient of non-saturating Loss:** What is the derivative of the non-saturating loss $\mathcal{L}_{NS}(\theta)$ with respect to $D_\phi(G_\theta(\mathbf{z}))$?

   Answer:

   $$\frac{\partial \mathcal{L}_{NS}}{\partial D_\phi\left(G_\theta\left(z\right)\right)} = -\frac{1}{D_\phi\left(G_\theta\left(z\right)\right)}.$$

   (d) **[4 points] Advantage of non-saturating loss:** Consider the derivatives of the saturating and non-saturating loss with respect to $D_\phi\left(G_\theta\left(\mathbf{z}\right)\right)$ at the beginning of training. Explain why non-saturating loss results in more stable generator training and reduces the saturation problem.

Hint: You might assume the generator produces samples that are easy to distinguish from real data at the beginning of training.

Answer: At the beginning of training, $D\left(G\left(\mathbf{z}\right)\right)$ is close to 0, and at the optimal discrminator and the optimal generator, $D\left(G\left(\mathbf{z}\right)\right)$ is 0.5. When $0 < D\left(G\left(\mathbf{z}\right)\right) < 0.5$, we see that $\|\frac{\partial \mathcal{L}_{NS}}{\partial D_\phi(G_\theta(z))}\| > \|\frac{\partial \mathcal{L}_{S}}{\partial D_\phi(G_\theta(z))}\|$.

$$x < \frac{1}{2} \Rightarrow 2x < 1 \Rightarrow x < 1 - x \Rightarrow \frac{1}{1-x} < \frac{1}{x},$$

where $x = D\left(G\left(\mathbf{z}\right)\right)$. The non-saturating loss has larger gradients (in magnitude) in the beginning. With this new gradient information, the generator trains faster, and the training process becomes more stable.

7. **[16 points total] Joint Temperature Scaling:**

   In problem set 1, you were asked to implement temperature scaling, a commonly used technique to calibrate the likelihood of the next token in language models. We showed that temperature scaling the logits of individual variables is different from temperature scaling the logits of the joint distribution.

   Let's consider a joint distribution $p(\mathbf{x})$ parameterized by an autoregressive model, where $\mathbf{x} = (x_1, x_2, \dots x_n)$ is a collection of discrete random variables. We define joint temperature scaling by a scalar $T > 0$ as the following:

$$\log p^T(\mathbf{x}) = \frac{1}{T} \log p(\mathbf{x}) - \log Z(T)$$

   where $p^T(\mathbf{x})$ is our temperature scaled joint distribution of interest and $Z(T)$ is the partition function.

   (a) **[2 points] Partition function:** What is the value of $Z(1)$, the partition function when $T = 1$?
   Answer: $Z(1) = 1$ because the original AR model is normalized.

   (b) **[2 points] Partition function:** Is the partition function $Z(2)$ for temperature $T = 2$ tractable to compute when n is large?
   Answer: It's intractable to compute for $T \neq 1$, runtime is exponential in n

   (c) **[2 points] Temperature effect:** Suppose we observe that samples $\mathbf{x}$ that have higher likelihood under $p(\mathbf{x})$ are judged to have higher quality by human evaluators. How should we set T to have p^T generate higher quality samples (compared to p)?
   Answer: Set T to be smaller than 1.

   (d) **[2 points] Sampling from $p^T(\mathbf{x})$:** Can we tractably sample from $p(\mathbf{x})$? Can we tractably sample from $p^T(\mathbf{x})$ (for an arbitrary T)?
   Answer: Yes for p, and no for p^T (unless $T = 1$ or $T = \infty$).

   (e) **[2 points] Learning from $p^T(\mathbf{x})$:** Now suppose we want to train another autoregressive model $q_\theta(\mathbf{x})$ to learn the distribution of p^T. Is there an unbiased Monte Carlo estimate of the following KL divergence

$$\mathcal{D}_{KL}(p^T(\mathbf{x}) \| q_\theta(\mathbf{x}))$$

   that we can evaluate tractably?
   Answer: No for forward, because we can't sample from p^T (and importance weights are intractable).

(f) **[4 points] KL Objective:** Let's examine the KL more closely.

Show that finding the q_θ that minimizes $\mathcal{D}_{KL}(p^T(\mathbf{x})\|q_\theta(\mathbf{x}))$ is equivalent to finding the q_θ that minimizes the following:

$$\mathbb{E}_{\mathbf{x}\sim p}\left[-e^{\frac{1-T}{T}\log p(\mathbf{x})}\log q_\theta(\mathbf{x})\right] \tag{$\star$}$$

(Hint: Importance sampling.)

Answer:

$$\begin{aligned}
\min_{q_\theta} \mathrm{KL}(p^T\|q_\theta) &= \min_{q_\theta} \mathbb{E}_{\mathbf{x}\sim p^T}\left[\log p^T(\mathbf{x}) - \log q_\theta(\mathbf{x})\right] \\
&= \min_{q_\theta} \mathbb{E}_{\mathbf{x}\sim p^T}\left[-\log q_\theta(\mathbf{x})\right] \\
&= \min_{q_\theta} \mathbb{E}_{\mathbf{x}\sim p}\left[-\frac{p^T(\mathbf{x})}{p(\mathbf{x})}\log q_\theta(\mathbf{x})\right] \\
&= \min_{q_\theta} \mathbb{E}_{\mathbf{x}\sim p}\left[-\frac{e^{\log p(\mathbf{x})/T - \log Z(T)}}{p(\mathbf{x})}\log q_\theta(\mathbf{x})\right] \\
&= \min_{q_\theta} \mathbb{E}_{\mathbf{x}\sim p}\left[-e^{\frac{1-T}{T}\log p(\mathbf{x}) - \log Z(T)}\log q_\theta(\mathbf{x})\right] \\
&= \min_{q_\theta} \mathbb{E}_{\mathbf{x}\sim p}\left[-e^{\frac{1-T}{T}\log p(\mathbf{x})}\log q_\theta(\mathbf{x})\right]
\end{aligned}$$

(g) **[2 points]** Is there an unbiased estimator of Equation ($\star$) that we can evaluate tractably?
Answer: Yes, since we can sample from p, and both p and q_θ have tractable likelihood.