STAT 453: Final Exam Study Bank

Exam Policy: Open-notes. NO ELECTRONICS / AI. Printed/handwritten notes allowed; no phones, laptops, tablets, smartwatches, calculators, earbuds, or AI tools; no internet or code execution.

1. **True/False — 1 pt each**

Statement	**Answer (T / F)**
1. Discriminative models parameterize and learn $p(y \mid x)$ directly.	T
2. Generative models can sample new x values, but cannot estimate $p(y \mid x)$.	F
3. Generative models are better at small sample sizes, while discriminative models are better at large sample sizes.	T
3. Because ReLU is two linear functions connected, models built with ReLU are always just linear models.	F
4. For generative models, a lower held-out negative log-likelihood (NLL) always implies better perceptual sample quality (e.g., FID).	F
5. Regularization (dropout, weight decay, early stopping) reduces overfitting by trading a bit of bias for lower variance.	T
6. Dropout is active during *evaluation* (test) mode.	F
7. The ReLU derivative is 0 for negative inputs.	T
8. Since the ReLU derivative is 0 for many values of pre-activation, many variants of the activation function have been developed to have non-zero slope everywhere.	T
9. Normalization doesn't matter for deep neural networks because there are many layers to make up for uncentered features.	F
10. BatchNorm uses running mean/variance at test time to normalize hidden activations.	T
11. Vanilla RNNs suffer from exploding / vanishing gradients because the reward signal must be back-propagated across many steps of multiplication.	T
12. LSTMs improve the exploding / vanishing gradient problem by having two parallel paths for cell state to be updated across steps of the sequence.	T
13. Vanilla RNNs use self-attention for long-range dependencies.	F

Statement (cont.)	**Answer (T / F)**
14. Because GANs train two coupled models (the discriminator and generator), they tend to be easier to train and less sensitive to hyperparamter choice than VAEs.	F
15. LLM "pre-training" is based on likelihood of tokens, not truthfulness of response.	T
16. LLM "post-training" uses reward signals that may be very distant from the predictions, leading to heavy reliance on reinforcement learning solutions.	T
17. Reinforcement Learning from Human Feedback (RLHF) almost always reduces hallucination rates because humans value and can evaluate truthfulness.	F
18. The *variational* in Variational AutoEncoders refers to making multiple predictions per input.	F
19. The VAE ELBO includes a KL-divergence term that pushes $q_\phi(z \mid x)$ toward the prior $p(z)$.	T
20. In DDPMs (diffusion models), the forward process is fixed Gaussian noising; the reverse process is learned.	T
21. Dropping a sensitive feature like 'race' from training features guarantees race-invariant predictions from a trained neural network.	F
22. CNNs provide rotation invariance by default without any augmentation or specialized layers.	F
23. Early stopping acts like an implicit regularizer (ridge-like in linear settings).	T
24. Deep learning models tend to be initialized toward more complicated functions, and learn over time to represent simpler functions.	F
25. The choice between discriminative and generative modeling depends on assumptions and prior knowledge (data regime, misspecification risk), not a universal rule.	T
26. The "Attention Is All You Need" paper showed that you do not need transformers, convolutions, or recurrences to build the best deep learning models.	F

2. **Matching – 2 pts each.** Each property is used **exactly once**. Write the **number** of the matching property next to each **algorithm / concept**.

Properties

1. Uses learned *keys* and *queries* to unlock *values* that define weighting.
2. Converges on separable data with a mistake bound depending on the margin.
3. Uses a smooth, differentiable activation that provides usable gradients, enabling gradient-descent training instead of discrete mistake updates.
4. Uses fully parallel sequence processing via attention, allowing direct content-based interactions between any two positions without recurrence.
5. Reduces overconfidence by assigning small probability mass to non-true classes; often improves calibration.
6. Normalizes layer activations using mini-batch statistics and learned scale/shift parameters.
7. Randomly masks activations during training; use inference-time scaling to match expected activations.
8. For a single example, $\nabla_z \ell = \hat{p} - y$ (predicted probs minus one-hot target).
9. Represents prediction penalty under a Normal (Gaussian) distribution of uncertainty.
10. Mitigates exploding gradients by limiting a parameter or gradient norm.

Algorithm / Concept	**Property (Number)**
1. Rosenblatt's Perceptron with Threshold activation	2
2. Rosenblatt's Perceptron with Sigmoid activation	3
3. Transformer architecture	4
4. Label smoothing	5
5. Batch Normalization	6
6. Dropout	7
7. Softmax + cross-entropy gradient	8
8. Gradient clipping	10
9. Self-attention	1
10. Squared-error loss	9

Multiple Choice – 3 points each subquestion

3. **Neural Net History**

   (a) **Rosenblatt's Perceptron.** Which statement is most accurate?

   - A. The perceptron converges on any dataset if you train long enough.
   - B. The perceptron implements a nonlinear decision boundary via a sigmoid.
   - [✓ C.] The perceptron converges on *linearly separable* data; it cannot represent XOR with a single layer.
   - D. The perceptron was introduced alongside backpropagation.

   (b) **Perceptron limitations (Minsky & Papert, 1969).** What was the core critique of Minsky and Papert?

   - A. Neural networks cannot be trained with gradient descent.
   - [✓ B.] Single-layer networks cannot represent certain simple functions (e.g., parity/XOR) without hidden units.
   - C. Multi-layer networks are less expressive than linear models.
   - D. Neural networks require exponential data for linear problems.

   (c) **Logistic regression / sigmoid units (1970s–1980s).** Which statement best distinguishes them from the original perceptron?

   - A. Logistic regression uses zero–one loss; perceptron uses cross-entropy.
   - [✓ B.] Logistic regression/sigmoid neurons optimize a smooth log-likelihood (cross-entropy), enabling gradient-based learning and probabilistic outputs.
   - C. Logistic regression requires labels in $\{-1, +1\}$; perceptron uses $\{0, 1\}$.
   - D. Sigmoid units eliminate the need for hidden layers.

   (d) **Chronology.** Which ordering is *earliest* → *latest*?

   - [✓ A.] PCA → LeNet → ResNet → Transformer
   - B. Transformer → ResNet → LeNet → PCA
   - C. ResNet → PCA → Transformer → LeNet
   - D. LeNet → PCA → Transformer → ResNet

4. **Statistical View of Deep Learning**

(a) **Gradient descent and MLE.** Which statement best explains the connection betwween MLE and gradient descent?

A. Gradient descent on any loss is equivalent to MLE.

[✓ B.] Minimizing the *negative log-likelihood* with (stochastic) gradient descent performs maximum likelihood estimation.

C. MLE requires closed-form solutions, so gradient descent is unrelated.

D. SGD is biased, so it cannot optimize likelihood-based objectives.

(b) **Regularization ↔ MAP.** Which pairing is conceptually correct?

[✓ A.] L2 weight penalty corresponds to a zero-mean Gaussian prior; L1 corresponds to a zero-mean Laplace prior.

B. L2 corresponds to a Laplace prior; L1 corresponds to a Gaussian prior.

C. Any penalty corresponds to a uniform prior.

D. Regularization only changes optimization speed, not the underlying statistical objective.

(c) **Data augmentation.** What statistical view best captures augmentation?

A. It increases the number of parameters to fit invariances explicitly.

[✓ B.] It encodes prior knowledge about $p(y \mid x)$ as invariances/equivariances, effectively performing MAP estimation.

C. It replaces likelihood with a margin loss, so it is non-probabilistic.

D. It guarantees calibration by smoothing logits at test time.

(d) **Early stopping.** Which explanation of early stopping is most accurate?

A. Early stopping reduces bias by allowing the model to fit noise.

[✓ B.] Early stopping limits the optimizer from fitting high-frequency/noise components; in linear settings it behaves like ridge (Tikhonov) regularization, improving the bias–variance trade-off.

C. Early stopping only changes training time and has no statistical effect.

D. Early stopping is equivalent to using a Laplace prior on the parameters.

5. **Optimization of Deep Models**

   (a) **Why first-order (concept).** Which is the *best* explanation for why deep learning predominantly uses first-order methods (SGD/Adam) rather than second-order (Newton/quasi-Newton) methods?

   A. Most loss functions are convex, so second-order are not needed.

   [✓ B.] Computing/storing/inverting (or even multiplying by) Hessians in high dimensions is prohibitively expensive.

   C. First-order methods tend to converge in fewer iterations (epochs) than second-order methods.

   D. GPUs cannot compute matrix–vector products with curvature information.

   (b) **Adam vs. SGD-momentum (practice).** Which comparison is most reasonable?

   [✓ A.] Adam typically excels with sparse/noisy gradients and faster initial convergence; SGD with momentum often yields stronger final generalization when tuned.

   B. Adam always converges faster and generalizes better than SGD.

   C. SGD with momentum is strictly better for all NLP tasks.

   D. Both are equivalent if the learning rate is the same.

   (c) **Warmup & schedules.** Why do large models (e.g., Transformers) often use LR warmup followed by a decay schedule (cosine/linear)?

   A. Warmup guarantees global optimality by avoiding saddle points.

   [✓ B.] Early steps stabilize training when parameters/normalization stats are poorly scaled; later decay trades speed for stability and generalization once near a good region.

   C. Decay increases the effective batch size without changing hardware.

   D. Warmup is only needed when using SGD but not Adam/AdamW.

   (d) **Gradient clipping.**Which statement is most accurate?

   A. Gradient clipping makes the loss strictly convex, guaranteeing convergence.

   [✓ B.] Clipping limits update magnitude to mitigate *exploding* gradients without changing the objective.

   C. Clipping primarily fixes *vanishing* gradients by amplifying small derivatives.

   D. Clipping works only with SGD and is incompatible with Adam/AdamW.

6. **Normalization and Initialization**

   (a) **BatchNorm at test time.** Which statement is correct?

   A. BN recomputes per-batch mean/variance on each single test example.

   [✓ B.] BN uses running (moving) estimates of mean/variance accumulated during training, with learned γ, β.

   C. BN freezes weights and disables affine parameters at test time.

   D. BN is identical to LayerNorm at test time.

   (b) **Small-batch finetuning.** Suppose you have a very small batch size (≤ 4). Which practice is *most* appropriate?

   A. Keep BN in training mode so running stats continue to adapt on tiny batches.

   [✓ B.] Freeze BN (eval mode using stored running stats) or replace with GroupNorm/LayerNorm to avoid noisy batch statistics.

   C. Increase learning rate so BN statistics track faster.

   D. Remove all normalization layers; initialization alone will suffice for stability.

7. **Discriminative vs. Generative Models**

   (a) Which statement best characterizes *discriminative* vs. *generative* models?

   [✓ A.] Discriminative models learn $p(y \mid x)$; generative models learn $p(x, y)$ (or $p(x \mid y)p(y)$).

   B. Discriminative models learn $p(x)$; generative models learn $p(y \mid x)$.

   C. Discriminative models simulate data; generative models cannot.

   D. Discriminative models require stronger distributional assumptions than generative models.

   (b) Which pair is a correct example of (discriminative, generative), respectively?

   [✓ A.] (Logistic regression, Naïve Bayes)

   B. (K-means, Softmax regression)

   C. (PCA, Kernel SVM)

   D. (GAN, Linear SVM)

   (c) In the low-data regime, which model often achieves lower error sooner *and why*?

   [✓ A.] Naïve Bayes, due to stronger modeling assumptions that reduce variance.

   B. Logistic regression, because it has strictly fewer parameters than Naïve Bayes.

   C. Logistic regression, because MLE is unbiased for all sample sizes.

   D. Naïve Bayes, because it maximizes the margin between classes.

(d) Which trade-off most directly explains the previous answer?

[✓ A.] Bias–variance trade-off.

B. Exploration–exploitation trade-off.

C. Precision–recall trade-off.

D. Depth–width trade-off.

8. **Practical Scenarios**

(a) **Video age-up filter (architecture).** You need a filter that makes people in a *video* look 50 years older while preserving identity and temporal consistency. Which setup is most appropriate?

A. An image-only style transfer network applied frame-by-frame with no temporal constraints.

B. A conditional ResNet classifier fine-tuned to predict age and then "invert" its logits to edit pixels.

[✓ C.] A conditional generative model (e.g., conditional diffusion or cGAN) with a *video-aware* backbone (3D U-Net / temporal attention) and identity/temporal consistency losses.

D. A k-means clustering of pixel colors followed by histogram equalization.

(b) **Video age-up training (data/objective).** Paired "same person now vs. +50 years later" videos are *not* available. What is a reasonable training strategy?

A. Supervised L2 loss between input and target frames using random elderly stock footage as targets.

[✓ B.] Unpaired translation or conditional generation with age conditioning (e.g., latent diffusion with age embeddings) plus identity loss (face-embedding consistency) and temporal consistency losses (e.g., optical flow / warping).

C. Train a face detector and use its bounding boxes as the supervision signal for aging.

D. Freeze a pre-trained classifier and fine-tune only BatchNorm statistics on elderly data.

(c) **Invariant predictions.** Stakeholders want predictions of a model to be *invariant to race*. You have access to race data, and you suspect proxies (variables correlated with race) exist in X. Which approach best targets the goal of having a predictive model invariant to race?

A. Drop the race column from X and proceed normally.

[✓ B.] In-processing *adversarial debiasing*: learn representations/predictions while an adversary (via gradient reversal) tries to recover race from the representation or logits; update to remove recoverable race signal.

C. Post-processing: calibrate probabilities on a validation set without using race.

D. Over-sample the minority racial group until classes are balanced.

(d) **Auditing via a GAN-style adversary.** You trained with adversarial debiasing. How can you *test* if race information still leaks from the learned representation $h(x)$ using only concepts covered in class?

A. Compute overall ROC–AUC; if high, leakage is impossible.

[✓ B.] Train a small *auxiliary discriminator* to predict race from $h(x)$ on a held-out set; high accuracy indicates residual leakage (mirrors the GAN discriminator idea).

C. Apply temperature scaling on logits until race cannot be predicted.

D. Add more weight decay and re-train the main model.

9. **Diffusion Models**

(a) What is the core idea behind diffusion (score-based) generative models?

[✓ A.] Define a fixed forward noising process that gradually destroys structure, then *learn the reverse denoising process* (often by predicting noise or the score $\nabla_x \log q(x_t)$) to iteratively transform Gaussian noise back into data samples.

B. Train a discriminator to classify real vs. fake images and update a generator to fool it in a single step.

C. Autoregressively predict the next token/pixel given the previous ones to sample in one left-to-right pass.

D. Encode data to a latent with a deterministic encoder and decode it back by minimizing only reconstruction loss.

(b) **Forward and reverse processes — choose one.** Which statement best describes the DDPM formulation?

[✓ A.] A fixed forward *Gaussian* Markov chain adds noise with schedule $\{\beta_t\}$ (so $q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$), and a learned reverse process models $p_\theta(x_{t-1} \mid x_t)$—often by predicting the added noise $\epsilon_\theta(x_t, t)$ to form the mean.

B. The forward process is learned, while the reverse is a known linear-Gaussian kernel requiring no training.

C. Both forward and reverse processes are deterministic; noise is used only for data augmentation.

D. The reverse conditional $p_\theta(x_{t-1} \mid x_t)$ is independent of timestep t once training converges.

(c) **Why predict noise ϵ instead of x_0 directly?**

[✓ A.] It yields a better-conditioned training signal across timesteps and aligns with denoising score matching (learning the score of $q(x_t)$).

B. It removes the need for a variance schedule and makes sampling deterministic.

C. It halves the number of sampling steps required at inference.

D. It guarantees lower FID than any x_0-prediction parameterization for fixed compute.

10. **CNNs**

(a) **Convolutions vs. fully-connected.** Which is *not* a core benefit of convolutions for images?

A. Parameter sharing across spatial locations

B. Translation equivariance

C. Local receptive fields

[✓ D.] Guaranteed rotation invariance without data augmentation

(b) **Receptive field.** Which change generally *increases* the effective receptive field *without* adding parameters?

A. Increasing kernel size from 3×3 to 5×5

[✓ B.] Using dilation (atrous) in convolutions

C. Adding a 1×1 convolution

D. Removing striding

(c) **Padding/stride.** A 3×3 conv with stride 2 and "valid" padding on a 32×32 feature map (assume just 1 input channel) outputs:

A. 32×32

B. 16×16

[✓ C.] 15×15

D. 14×14

11. **Autoencoders & Variational Autoencoders (VAEs)**

    (a) **Bottleneck.** Which statement best explains the purpose of a "bottleneck" (low-dimensional z) in standard autoencoders?

    A. To guarantee perfect reconstruction on any dataset.

    B. To make the decoder linear so training is convex.

    [✓ C.] To prevent a trivial identity mapping and force the model to learn a compressed, informative representation that generalizes.

    D. To eliminate the need for regularization or early stopping.

    (b) **Vanilla AE vs. VAE.** Which description is most accurate?

    A. Both use a deterministic encoder $z = f_\phi(x)$ and minimize only pixel MSE.

    [✓ B.] A vanilla AE learns a deterministic code and minimizes reconstruction loss; a VAE posits a *probabilistic* encoder $q_\phi(z \,|\, x)$ and decoder $p_\theta(x \,|\, z)$ with a prior $p(z)$, trained by maximizing an ELBO (reconstruction term + KL to the prior).

    C. A VAE removes the decoder and replaces it with a discriminator.

    D. The only difference is that VAEs always use larger latent dimensionality.

    (c) **ELBO (objective).** Which is the standard ELBO for a VAE with prior $p(z)$, decoder $p_\theta(x \mid z)$, and encoder $q_\phi(z \mid x)$?

    A. $\mathbb{E}_{p_{\text{data}}(x)}\big[\log p_\theta(x)\big]$

    B. $\mathbb{E}_{q_\phi(z)}\big[\log p_\theta(x \mid z)\big] - \mathrm{KL}(q_\phi(z) \,\|\, p(z))$

    [✓ C.] $\mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x \mid z)\big] - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))$

    D. $-\mathrm{KL}(p_\theta(x \mid z) \,\|\, q_\phi(z \mid x))$

    (d) **Posterior collapse — choose one.** Which intervention is most appropriate when $q_\phi(z \mid x)$ ignores x (the decoder does all the work)?

    [✓ A.] Apply *KL annealing / warm-up*, gradually increasing the KL weight so the encoder carries information before matching the prior strongly.

    B. Increase β (β-VAE with $\beta > 1$) to push $q_\phi(z \mid x)$ toward the prior more aggressively.

    C. Make the decoder more expressive and the encoder weaker to ease optimization.

    D. Increase decoder capacity and remove input noise/dropout to improve reconstruction fidelity.

12. **Sequence Models: RNNs → Attention**

(a) **Limitation of vanilla RNNs on long sequences.** Which statement best captures the limitation and how attention addresses it?

A. RNNs overfit small datasets; attention reduces parameters via weight sharing.

B. RNNs cannot model sequences longer than 512; attention extends the maximum length.

[✓ C.] RNNs suffer vanishing/attenuated gradients over long dependencies; attention creates direct, weighted connections between distant tokens and enables parallel computation.

D. RNNs require teacher forcing; attention removes exposure bias entirely.

(b) **Scaled dot-product attention.** Which option gives the correct formula *and* role of the $1/\sqrt{d_k}$ factor?

A. $\text{softmax}(QK^\top)\,V$; the scale increases gradient magnitude for faster training.

[✓ B.] $\text{softmax}\big(\frac{QK^\top}{\sqrt{d_k}}\big)V$; the scale keeps logits in a reasonable range as d_k grows, stabilizing the softmax and its gradients.

C. $\text{softmax}\big(\frac{KV^\top}{\sqrt{d_k}}\big)Q$; the scale prevents overfitting by shrinking parameters.

D. $\text{softmax}\big(\frac{QK}{d_k}\big)V$; the scale normalizes by sequence length.

(c) **Scaling to LLMs — choose one.** Which set of architectural features best explains Transformer scalability?

[✓ A.] Residual (skip) connections, multi-head attention, and layer normalization.

B. Batch Normalization in every sublayer, multi-head attention, and residual connections.

C. Residual connections, sparse MoE, and layer normalization were all part of the original Transformer blocks.

D. Multi-head attention alone; normalization and residuals are not needed for deep scaling.

13. **GANs**

(a) **Vanilla GAN objective (Goodfellow et al., 2014).** Which expression is the correct minimax objective?

[✓ A.] $\min_{G} \max_{D} \Big[\mathbb{E}_{x\sim p_{\text{data}}} \log D(x) \; + \; \mathbb{E}_{z\sim p(z)} \log\big(1 - D(G(z))\big) \Big]$

B. $\min_{G} \mathbb{E}_{z\sim p(z)}\big[-\log D(G(z))\big]$

C. $\min_{G} \max_{D} \Big[\mathbb{E}_{x\sim p_{\text{data}}} D(x) \; - \; \mathbb{E}_{z\sim p(z)} D\big(G(z)\big) \Big]$

D. $\min_{G} \max_{D} \Big[\| \phi(x) - \phi(G(z)) \|_2^2 \Big]$

(b) **Failures and mitigations.** Which pairing is *most* correct?

[✓ A.] Mode collapse → mini-batch discrimination or feature matching can encourage sample diversity.

B. Vanishing generator gradients → use the minimax (saturating) loss $\log(1 - D(G(z)))$ for G.

C. Lipschitz issues → remove all normalization layers from both G and D to stabilize dynamics.

D. Overpowerful discriminator early → increase D updates per G step and remove label smoothing.

(c) In the *non-saturating* GAN formulation, which generator objective is used to mitigate vanishing gradients early in training?

[✓ A.] $\min_{G} \mathbb{E}_{z\sim p(z)}\big[-\log D\big(G(z)\big)\big]$

B. $\min_{G} \mathbb{E}_{z\sim p(z)}\big[\log\big(1 - D\big(G(z)\big)\big]$

C. $\min_{G} \mathbb{E}_{z\sim p(z)}\big[-D\big(G(z)\big)\big]$

D. $\min_{G} \big\| \mathbb{E}_{x\sim p_{\text{data}}} \phi(x) - \mathbb{E}_{z} \phi(G(z)) \big\|_2^2$

14. **LLM Pretraining and Tokenization**

(a) **Objective.** What is the standard unsupervised pretraining objective for decoder-only LLMs?

A. Masked language modeling (random token masking)

B. Next-sentence prediction

[✓ C.] Next-token prediction with causal masking, trained by cross-entropy (MLE)

D. Denoising autoencoding of shuffled spans only

(b) **Why subword tokenization?**

A. It guarantees perfect word segmentation for all languages.

[✓ B.] It balances vocabulary size and sequence length, handles rare/novel words by composing from frequent subunits, and captures morphemes.

C. It eliminates the need for embeddings.

D. It makes training objective convex.

(c) **Training stability/efficiency.** Which practice bundle is most appropriate for large-scale pretraining?

[✓ A.] Learning-rate warmup then decay schedule (e.g., cosine), mixed-precision training (FP16/BF16), and gradient clipping.

B. Large constant learning rate with no decay, full precision only, and no clipping.

C. Warmup only (no decay), switch to full precision for stability, and increase batch size aggressively without tuning LR.

D. Cosine decay only (no warmup), disable clipping, and use per-parameter learning rates instead of a schedule.

15. **LLM**

(a) **Motivation.** Why use parameter-efficient fine-tuning (PEFT) methods like LoRA/adapters?

A. They avoid backpropagation entirely

[✓ B.] They reduce trainable parameters and memory/IO, enabling finetuning large models on modest hardware

C. They improve zero-shot accuracy without any finetuning data

D. They permanently change the base model weights during training

(b) **LoRA vs. full finetune.** Which statement is accurate?

[✓ A.] LoRA inserts trainable low-rank matrices into certain weight paths; at inference these can be merged with base weights

B. LoRA adds full-rank matrices to every layer norm

C. LoRA requires updating all original parameters with a smaller learning rate

D. LoRA prevents multi-task finetuning because adapters cannot be swapped

(c) **SFT vs. RLHF.** Which description is most accurate?

A. SFT learns from preference comparisons; RLHF learns from labeled input–output pairs.

B. Both SFT and RLHF optimize the same cross-entropy on the same data.

[✓ C.] SFT fine-tunes on paired demonstrations (inputs → target outputs); RLHF optimizes a policy using a learned reward model fit to human preferences.

D. RLHF is just weight decay applied after SFT.

(d) **Shift risk & evaluation.** Which pairing correctly states a risk under distribution shift *and* a reasonable evaluation?

A. Risk: catastrophic forgetting; Eval: measure bits-per-byte on the pretraining corpus.

[✓ B.] Risk: reward hacking/specification gaming; Eval: held-out domain preference tests and red-teaming for unintended behaviors.

C. Risk: memorize users' prompts; Eval: increase context window.

D. Risk: mode collapse; Eval: FID on image samples.

Short Answer – 5 points each

16. **Discriminative vs. Generative**

    (a) You train logistic regression for $p_\theta(y \mid x)$ and a Naïve Bayes model with class-conditional Gaussians. In the low-data regime, which would you expect to reach lower error sooner and why?

    (b) For deep generative models, does the above argument still hold? Why or why not?