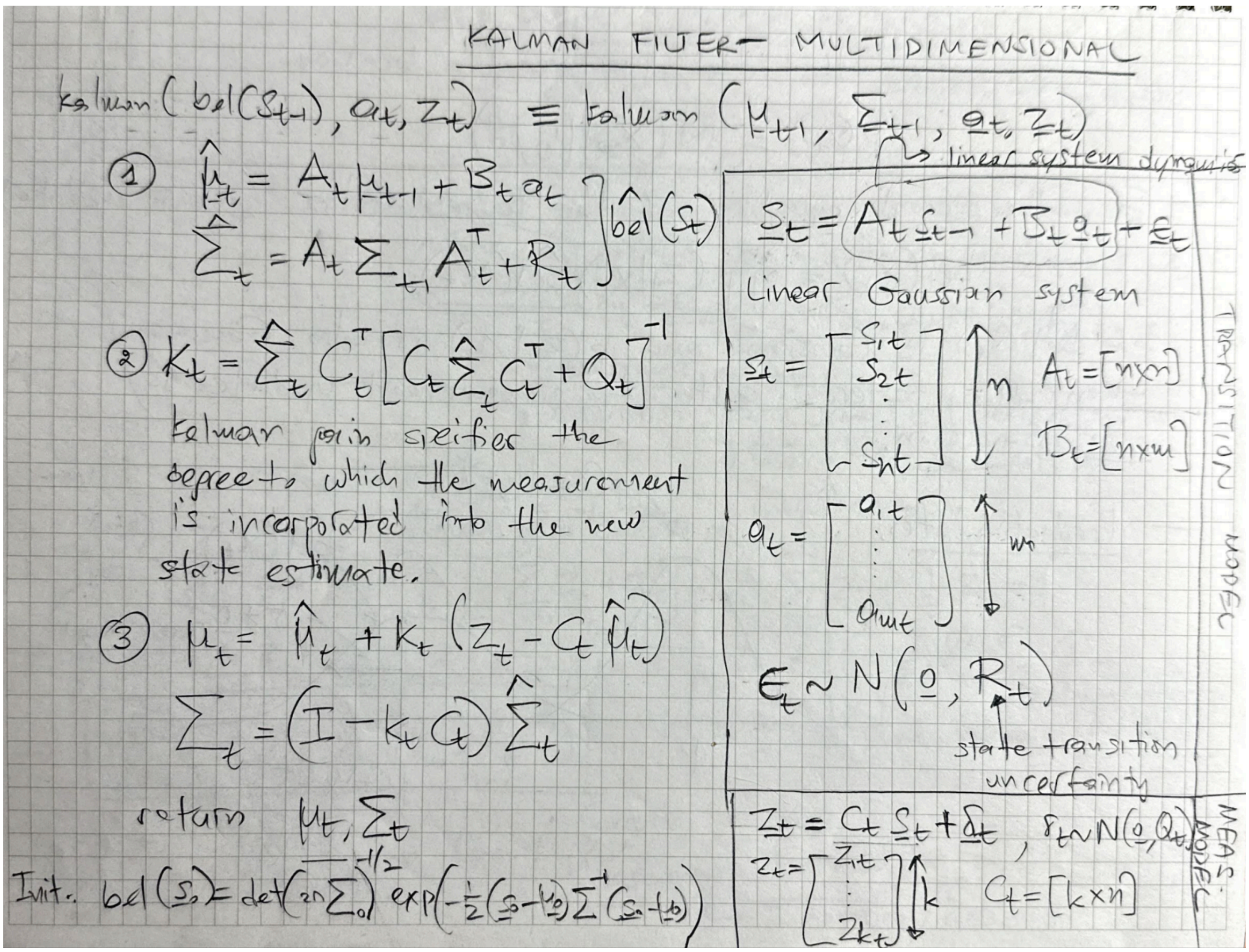
Introduction to AI

NYU Spring 2025 In-Person Final Exam - SOLUTIONS

QUESTION SET 1 (15 points)



Which of the following statements regarding the system described above is / are correct ?

A. If $C_t = \mathbf{0}, \forall t$, then it is not possible to compute an estimate of the state using the Kalman Filter.

B. The Kalman Filter gain K_t depends on the measurements $z_{1:t}$.

C. None of the provided answers is correct.

Answer:

A. If $C_t = 0$ then then no information about the state is ever observed—the measurements z_t contain no state-dependent signal. The Kalman filter needs observations to update state estimates; without them, it becomes a pure prediction step with no correction. Thus, a meaningful state estimate cannot be updated from observations.

Some students may have selected B. The Kalman gain K_t is a function of the covariance of the state estimate and the covariance of the measurement noise. It does not depend on the measurements $z_{1:t}$. Substract 7.5 points for this choice.

Some students may have selected C. Subtract 15 points.

QUESTION SET 2 (25 points)

QUESTION SET 2A (10 points)

In the following gridworld environment, the agent can move in four directions: south, west, east, north. The agent receives a reward of +1 for reaching the terminal state (1,4) and -1 for reaching the terminal state (2,4). The walls are represented by "wall" and the empty cells are represented by their coordinates.

```
      1      2      3      4
1   (1,1)  (1,2)  (1,3)  +1 (terminal)
2   (2,1)   wall  (2,3)  -1 (terminal)
3   (3,1)  (3,2)  (3,3)  (3,4)
```

What is the number of possible deterministic policies for this grid-world? Explain your answer.

Answer:

The number of deterministic policies in a grid world is:

$$|\Pi| = |A|^{|S|}$$

where:

- $|A|$ is the number of available actions
- $|S|$ is the number of non-terminal states in the grid.

In a 3×4 grid world there are 2 terminal states and 1 wall (impassable), that leaves:

- $3 \times 4 = 12$ total cells
- $|S| = 12 - 2\ (\text{terminal}) - 1\ (\text{wall}) = 9$

So the number of deterministic policies is:

$$|\Pi| = 4^9 = 262{,}144$$

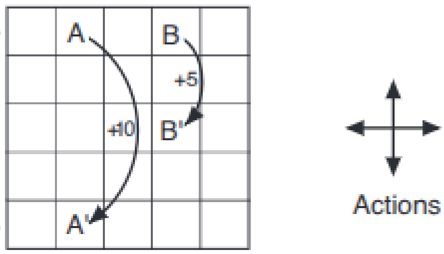
Some students may have not provided the exact number but if the formula is correct dont subtract points.

Some students may have answered erroneously that:

$$|\Pi| = |S|^{|A|}$$

Subtract 5 points in this case.

QUESTION SET 2B (15 points)



3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

In the above figure, the gridworld environment is a 5x5 grid that is deterministic. - The agent can move in 4 directions: North, South, East, West.

- The agent operates under a uniform random policy — the probability of each action at each state is 0.25.
- If a move would take the agent off the grid, it remains in the same state.
- Actions that would take the agent into a wall result in a -1 reward and the agent remains in the same state. Other actions result in a reward of 0 indepedent of the state unless the agent is in a special state as described below.
- Discount factor: $\gamma = 0.9$ - this is just a value used to produce the numbers shown in the figure.
- Two special states:
  - A: when the agent is in state A and chooses any action, it is instantly transported to state A' = A → teleport to state A' with reward 10.
  - B: when the agent is in state B and chooses any action, it is transported to B' = B → teleport to state B' with reward 5.

The value function $v(s)$ for all states under the random policy, using the Bellman equation for policy evaluation is given by:

$$v(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \left[r(s,a,s') + \gamma v(s') \right]$$

Explain using the equation (this is not a numerical exercise) why the value of state A is less than its immediate reward of 10 while the value of state B is greater than its immediate reward of 5.

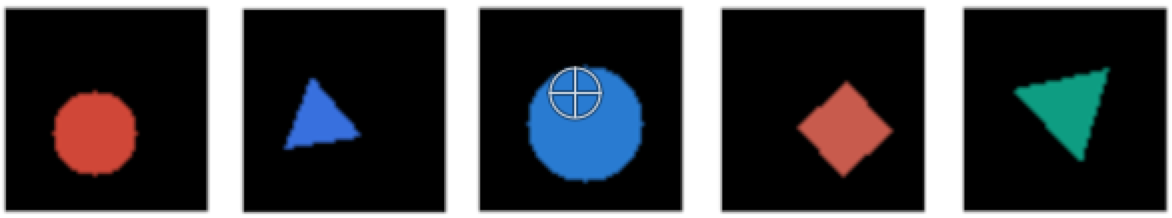
Answer:

The agent that is teleported to state A' receives an immediate reward of 10. However, the value of state A is less than 10 because the agent from the teleported state can run into a wall with some probability and this is reflected by the value of A’. This does not happen in the teleported state B’ that has a positive value.

QUESTION SET 3 (25 points)

You are given images of geometrical shapes and description of the images in natural language e.g for the shapes below these are:

“red circle”, “blue triangle”, “blue circle”, “red square”, “gren triangle”

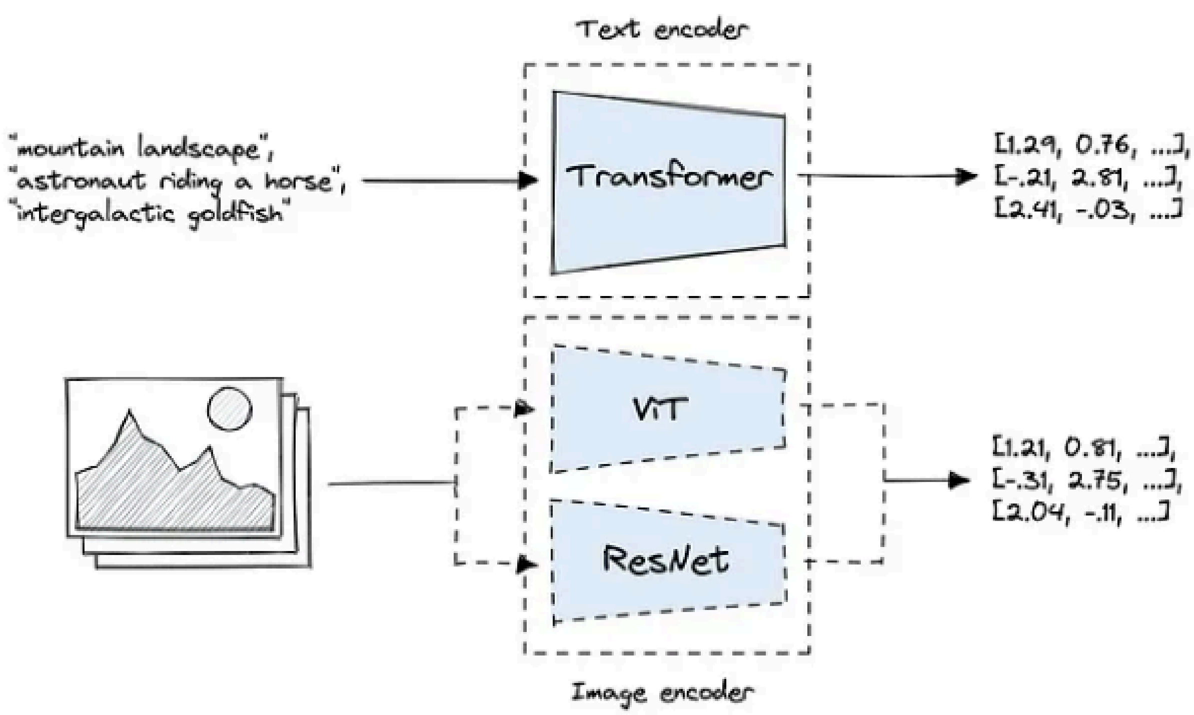


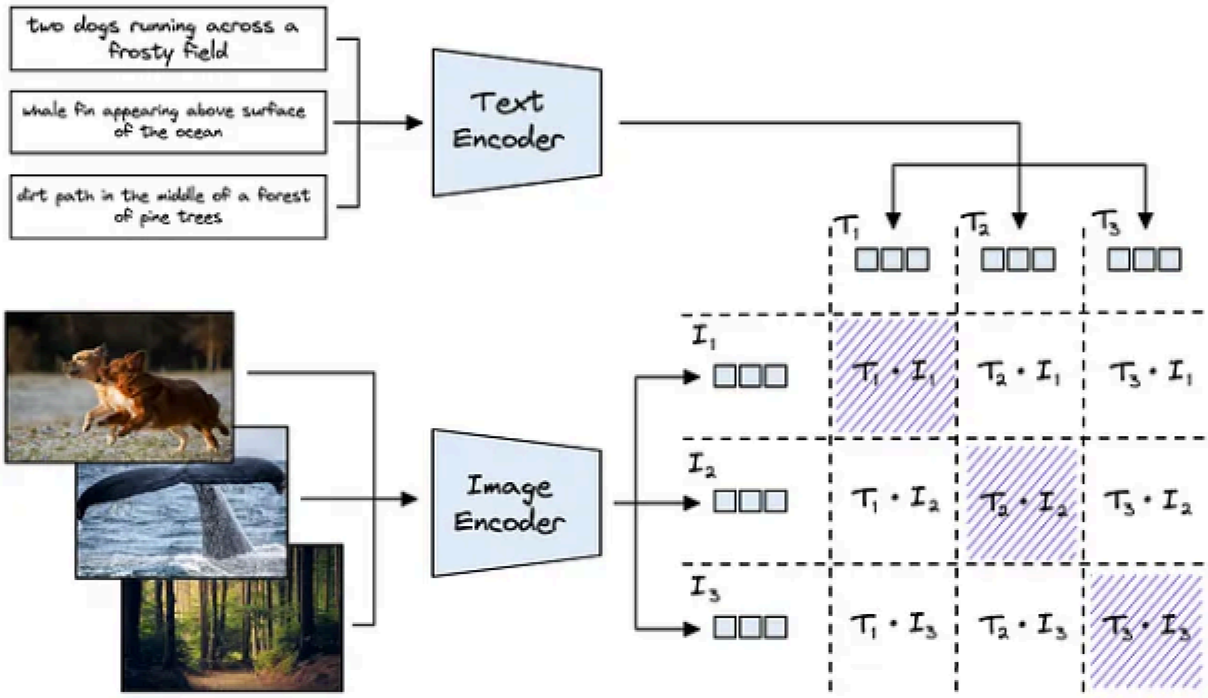
You can assume you have a dataset of size $h \times w$ and their aligned descriptions of size s tokens.

QS3A (15 points)

Draw an architecture of a system that can be used to learn the mapping between the images and the descriptions.

Answer:





Students must specify either a ResNet/CNN or a ViT as the image encoder. The image encoder must be followed by a projection layer to project the image features into the same space as the text features. Its imperative to be granted all points to check that the similarity of the image and text features is computed using a dot product.

QS3B (10 points)

Explain clearly the equation of the loss function and the training process.

Answer: The loss function has two components L^+ and L^- and the total loss is the sum of the positive and negative components. The positive component encourages the model to learn a joint embedding space where similar images and texts are close together, while the negative component encourages dissimilar images and texts to be far apart.

Students may have offered equations that are associated with CE or MSE or other loss functions but the essence of capturing all points is the recognition that the loss function has these two components that have the two specified roles above. Award points the closer the answer is to the one above.

QUESTION SET 4 (20 points)

Explain the advantages of llama3-8B over llama2-7B for each of the following features. `B` stands for billion parameters.

Feature	LLaMA 2	LLaMA 3
Model Sizes	7B, 13B, 70B	8B, 70B, 405B
Layers	32 (7B), 40 (13B), 80 (70B)	32 (8B), 80 (70B), 126 (405B)
Latent Size	4096 (7B), 5120 (13B), 8192 (70B)	4096 (8B), 8192 (70B), 16,384 (405B)
Attention Heads	32 (7B), 40 (13B), 64 (70B)	32 (8B), 64 (70B), 128 (405B)
FF Dimension	11,008 (7B), 13,312 (13B), 22,528 (70B)	14,336 (8B), 28,672 (70B), 53,248 (405B)
Vocabulary Size	32,000	128,000
Context Length	4,096 tokens	8,192 tokens
Training Tokens	~2 trillion	~15 trillion

HINT: Ensure that you quote any interdependencies between the features. Any explanation that argues “bigger is better” will not be accepted.

Answer:

Latent size: A larger latent size increases the representational capacity of each token embedding. This leads to richer feature representations, especially in combination with larger FFN dimensions.

FF Dimension: A higher FFN dimension increases the capacity of the intermediate transformation layer between attention blocks.

Vocab size: Vocabulary reduces out-of-vocabulary errors and supports multilingual and multi-domain (eg coding vs not) understanding better. It also improves tokenization efficiency (fewer tokens per input).

Context length: A larger context length allows the model to process longer sequences of text, which is particularly useful for tasks requiring long-term dependencies or understanding of larger contexts. Coupled with the improved token efficiency llama3 offered even longer effective contexts relative to llama2 than the pure ration indicates.

Training tokens: A significantly larger number leads to better generalization.

QUESTION SET 5 (15 points)

You are asked to reduce computational complexity of solving the MDP by selecting the γ discount factor. Explain why this is a good idea and how you will select the γ value to effectively reduce the computation associated with estimating the $Q(s, a)$ value function.

Answer:

The discount factor γ controls the effective planning horizon of the agent.

A smaller γ places more weight on immediate rewards and discounts distant future rewards more heavily. As a result, the expected return $Q(s, a)$ converges faster, and the Bellman updates require fewer steps of value propagation across the state space. This limits the depth of the backup tree, which directly reduces computational effort, especially in dynamic programming.

PS: On the other hand a larger γ captures long term rewards that may be important in some applications, creating a trade-off between computational efficiency and performance fidelity. For example when facing problems like computer games that typically have long horizons, then larger gamma values are needed.