

# LLM Sys

## 11868 LLM Systems Decoding

Lei Li



**Carnegie Mellon University**

**Language Technologies Institute**

# Recap about Tokenization

- Subword tokenization: Byte-Pair-Encoding
  - iteratively merging most frequent pairs of tokens
- Information-theoretic vocabulary (VOLT)
  - solving entropy constrained optimal transport problem
- Pre-tokenization through regex
- Number treatment
- Vocab sharing impact multilingual performance
  - how to solve languages in stagnant quad

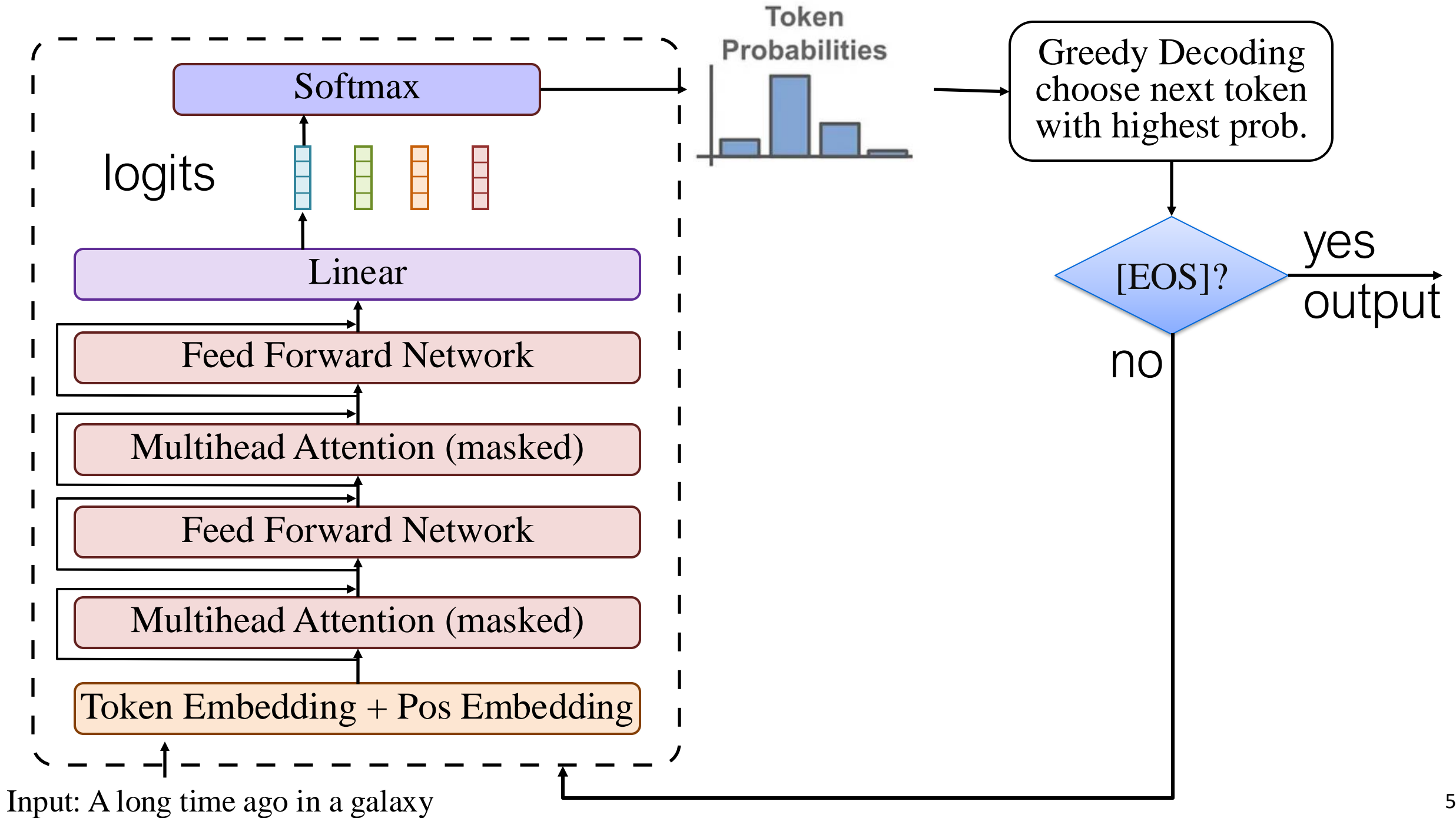
# Outline

- How to generate text at inference time

# Sequence Decoding

$$\operatorname{argmax}_y P(y|x) = f_{\theta}(x, y)$$

- naive solution: exhaustive search
  - too expensive  $O(V^N)$
- Greedy (max) decoding
- Sampling
- Beam search
  - (approximate) dynamic programming



# Max Decoding

- For every next token, pick the one that maximizes the probability

$$\max p(x_t | x_{1 \dots t-1})$$

- equivalent to maximizing logits, no need to normalize

# Sampling

- Instead of  $\operatorname{argmax}_y P(y|x) = f_\theta(x, y)$
- Generate samples of translation  $Y$  from the distribution  $P(Y|X)$
- Q: how to generate samples from a discrete distribution?

# Discrete Sampling

- sample  $n$  values  $x$ 's from  $k$  categories, with prob.  $p_1, p_2, \dots, p_k$
- Direct sampling:  $O(nk)$
- Binary Search:  $O(k + n \log k)$
- Alias sampling:  $O(k \log k + n)$

```
probs = torch.softmax(logits, dim=-1)
next_token = torch.multinomial(probs, num_samples=1)
```



# Fast Sampling with Gumbel Max Trick

- sampling from  $\text{Categorical}(\text{Softmax}(h))$  is equivalent to

$$\arg \max x$$

$$z \sim \text{Uniform}(0,1)$$

$$x = h - \log(-\log z)$$

- Theory:  $x$  follows Gumbel distribution, and  $\arg \max x$  follows

$$\text{Categorical}\left(\frac{\exp h_i}{\sum_{j=1}^k h_j}\right)$$

```
class GumbelSampler:
    def __init__(self, batch_size, vocab_size, device):
        self.batch_size = batch_size
        self.vocab_size = vocab_size
        # Pre-compute noise
        self.noise = self._prepare_gumbel_noise(device)

    def _prepare_gumbel_noise(self, device):
        # Generate noise tensor once
        uniform_noise = torch.rand(self.batch_size,
self.vocab_size, device=device)
        return -torch.log(-torch.log(uniform_noise))

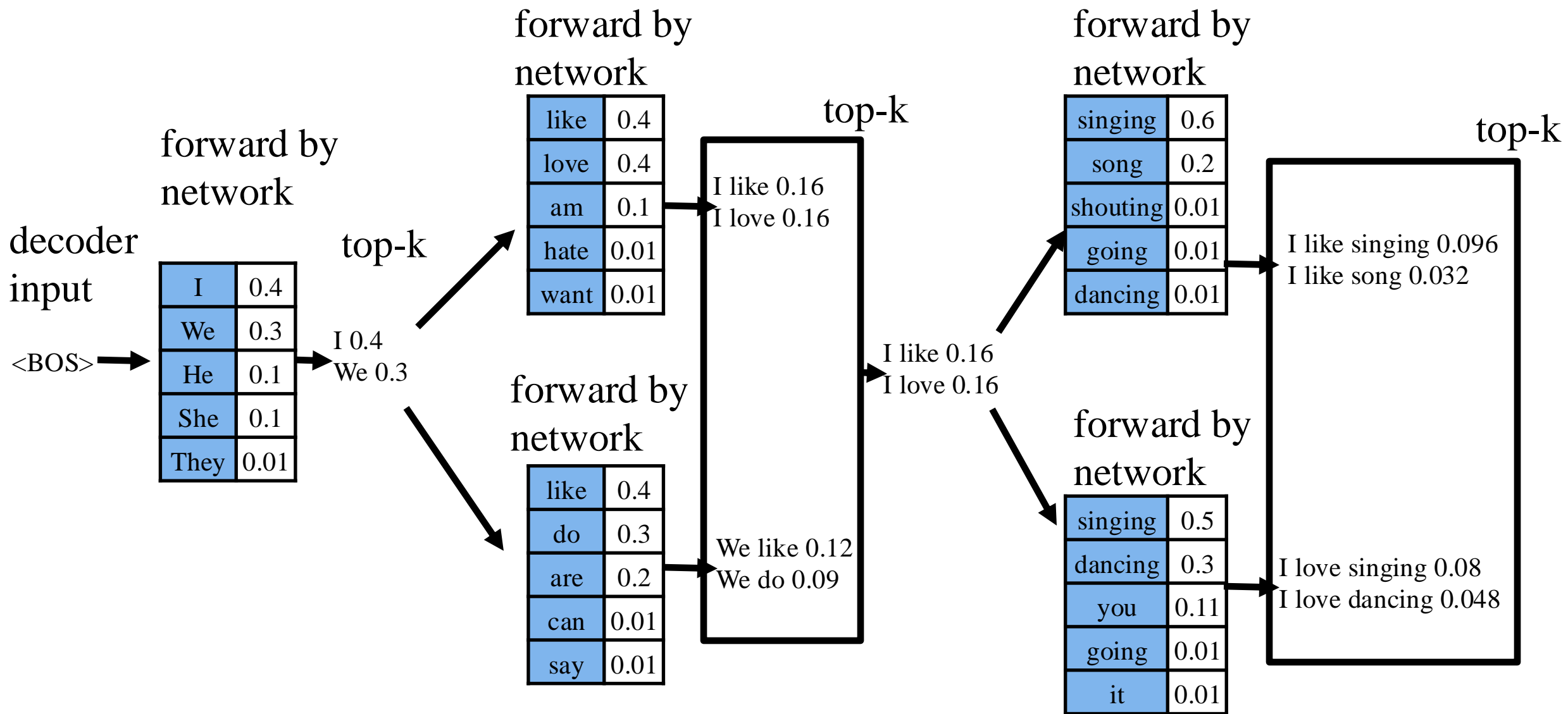
    def sample(self, logits):
        # Direct sampling without softmax
        return torch.argmax(logits + self.noise, dim=-1)
```

# Beam Search

Find approximate solutions to  $\operatorname{argmax}_y P(y|\mathbf{x}) = f_{\theta}(x, y)$

1. start with empty S
2. at each step, keep k best partial sequences
3. expand them with one more forward generation
4. collect new partial results and keep top-k

# Beam Search



```

best_scores = []
add {[0], 0.0} to best_scores # 0 is for beginning of sentence token
for i in 1 to max_length:
  new_seqs = PriorityQueue()
  for (candidate, s) in best_scores:
    if candidate[-1] is EOS:
      prob = all -inf
      prob[EOS] = 0
    else:
      prob = using model to take candidate and compute next token probabilities (logp)
      pick top k scores from prob, and their index
      for each score, index in the top-k of prob:
        new_candidate = candidate.append(index)
        new_score = s + score
        if not new_seqs.full():
          add (new_candidate, new_score) to new_seqs
    else:
      if new_seqs.queue[0][1] < new_score:
        new_seqs.get() # pop the one with lowest score
        add (new_candidate, new_score) to new_seqs

```

# Pruning for Beam Search

- Relative threshold pruning
  - prune candidates with too low score from the top one
  - Given a pruning threshold  $r_p$  and an active candidate list  $C$ , a candidate  $\text{cand} \in C$  is discarded if:  $\text{score}(\text{cand}) \leq r_p * \max\{\text{score}(c)\}$
- Absolute threshold pruning:
  - $\text{score}(\text{cand}) \leq \max\{\text{score}(c)\} - a_p$
- Relative local threshold pruning

# Combine Sample and Beam Search

- Sample the first tokens
- continue beam search for the later
- why?

# Code example

- [https://github.com/lmsystem/lmsys\\_code\\_examples/blob/main/decoding/decoding.ipynb](https://github.com/lmsystem/lmsys_code_examples/blob/main/decoding/decoding.ipynb)



# HW1 Survey

[https://docs.google.com/forms/d/1btWd65-BZJLrnofPkMYaEWqb7N6Bgi3AoSKBKhuggXY/viewform?edit\\_requested=true](https://docs.google.com/forms/d/1btWd65-BZJLrnofPkMYaEWqb7N6Bgi3AoSKBKhuggXY/viewform?edit_requested=true)

# Project

- <https://lmsystem.github.io/lmsystem2024spring/docs/Projects>
- Proposal due: 2/26
  - You are highly encouraged to discuss your project with TAs
- Mid term Report: 4/1
- Poster Project Presentation: 4/24 or 4/25 (depending on room availability)
- Final Report: next day

# Project Proposal

- What LLM System problem are you planning to address?
  - what are the **system challenges**?
- What are the existing state-of-art methods on this problem? Is the source code/model available?
- Possible directions for going forward.
- How do you evaluate the performance? what kind of workload?
- Who is your team and how are you planning to split the workload between team members?
- A rough timeline/milestones
- What CPU, GPU and storage infrastructure do need for this project? Please estimate the amount of computation time required.

# Project Report Requirement

- Introduction/Motivation: This essentially lays out the problem definition, motivation, talks about why we need to work on it, the key contributions expected/presented in the work.
- Related Work/Background: This talks about key papers/works that provide context to your current work. Instead of listing down multiple past works, talk about the ones that minimally differ from your work, and how.
- Methodology: This section talks about your method, raises research questions and how you are going to address them.
- Experiments: This section can describe your experiments and the results you obtain.
- Analysis/Ablations: Typically, you would have multiple factors involved in your experimental setting. Analysis sections help you probe deeper into the results and help piece out contributions from individual modeling decisions made.
- Conclusion/Discussion: This would list the main takeaways from your work, discuss some future ideas (if any) and engage in discussion.
- Limitations: This section lays out some known limitations of your work.
- [final report only] Team Member Contributions List out each individual's contributions in this section.

# Project Team Pairing