Proposal Guidelines

For the structure of the proposal, we expect you can follow a research paper format, which consists of introduction, related work, datasets/environments, methodology, and evaluation description. In general, we want to see that you have a clear goal in the project. The technical details can be described in a rough manner, but in principle, you need to show what problem you want to study.

- Introduction: This part includes the background and motivation of your project. The problem has to be well-defined. What are the input and output. Why this is an important problem to study.

- Related work: Every research work must include a description of the current state of the problem. In this part describe what has been done in previous work on the same or related subject. We don't want you to just download their source code and run it on the released data. We want you to do something interesting. So here describe why what you propose to do here is useful, any aspect is novel and different from prior work. Doing comparison work can also be treated as novel if nobody has reached the same level of depth or scale.

- Datasets/Environments: For problems that work on a dataset, the following information is required. What data do you want to use? What is the size of it? What information is contained? Why is it suitable for your task? For problems that involve interactions with an environment: what is the mechanism of this environment? What is the complexity/total number of states of this environment? What is the input/output structure of an interaction?

- Methodology: What models do you want to use? You may change the model as the project goes, but you may want to indicate some type of models that might be suitable for your problem. For example, is it a supervised learning problem or unsupervised? What classifiers can you start with? Are you making improvements? Are you going to implement any baseline methods? You don't have to be crystal clear on this section, but it can be used to indicate the direction that your project goes.

- Evaluation: What metrics or methods do you want to use for evaluating your models?

Length: 1 page (or more if necessary). Single space if MS word is used. Or you can choose a latex template for a CS conference, e.g., https://www.acm.org/publications/proceedings-template.

Grading: Based on each section described above, 20 points per section. As you can tell, they're related to each other. Each group member will make separate submission with all group members' names indicated.

Progress Report Guidelines

For the progress report, you are asked to provide descriptions on the current status of your project and your future plan. These may include the following contents:

- **Data preprocessing/deep dive into the environment.** If you are working on a dataset, how did you preprocess the dataset? Which features do you select, and for which purposes? What is the resulting dataset (size, splits, any characteristics that are relevant to the project)? If you are working on an environment, include your analysis and approach on this environment. I.e., what are the challenging parts in this environment. What is your formulation of the problem (e.g., state/action representation)? Make sure you motivate the decisions made along the way.

- **Methodology.** Report what are your current models/methods in solving the problem. Including: model architecture, parameter spaces, how to train the model, any model selection and hyperparameter tuning. Do you use any comparisons? What are the baselines (if applicable)? What is the rationale for designing such a model (your proposal)?

- **Current result.** Report the results you've obtained thus far. Describe the evaluation metrics or objectives/goals. Including: what are the performances of your model on the problem and the baseline methods if applicable.

- **Future plan for improvement.** Report what you are anticipating to improve and achieve in the finalized version.

Length: 2 pages (or more if necessary). Single space if MS word is used. Or you can choose latex templates, e.g. https://www.acm.org/publications/proceedings-template.

Grading: Based on each section described above, 25 points per section. As you can tell, they're related to each other. Each group member will make a separate submission with all group members' names indicated.

Final Report Guidelines

Problem Description

- What is the task?
    - A description of the particular problem(s) you are addressing
    - The motivation behind the project and why this is an important problem to study (e.g. you can discuss its social impact or other impact).
- System input and output (Examples are encouraged)
    - What is the input that the model receives?
    - What is the output that the model generates?
- What are the challenges of the task?
- What are your contributions in the project? Give a brief overview of what you have done and achieved. Report the task allocations among your team members

Reference/Related work

- Put your work in context: What has been done before related to your topic?
    - Please make sure that you cite relevant and (if possible) relatively new research.
- What's new in your work? E.g. Do you propose a new model or model variant? Do you provide new analysis or insights which can be used for future model development?

Methodology

- What are your datasets?
    - What data do you use?
    - What is the size of it?
    - What information is contained? (Examples are encouraged)
    - Why is it suitable for your task?
    - Provide some details about the dataset (e.g. If it is a classification task, is the dataset biased? If so, what is the percentage of each class? Are there any characteristics of the data?)
- Preprocessing of the data (if needed)
    - How do you collect and clean the data?
    - How do you process the data to fit into the structure that the model expects?
- What are your environments?
    - What environments do you choose?
    - What are the complexities of them?
    - What are the interaction mechanisms?
    - What are the input-output structures?
    - What are the goals of these environments?
- What methods do you experiment with? And why do you think they're reasonable and suitable for the task? Please include the details of your model architecture (e.g. for a neural model, number of layers, loss functions, parameters, etc.) A diagram for your model can be helpful here.

Experiments

- What is your dataset size? How do you split the dataset into training/validation/test sets? Please also include the implementation details of the training process of your model (learning rate, batch size, number of epochs, parameter tuning, etc.)
- What evaluation metrics do you use? Are they proper to calibrate system performance? Have a discussion on pros and cons if needed.
- What are your models and what are their performances?
    - What are the baselines and what are their performances? How do they compare with your model?
    - Discuss your results. If it is not satisfying, what might be the reasons? Have you tried other variants?
    - Summarize the insights you have gained through this exercise. You can do an error analysis or other types of analysis here.

Note: Please use tables, figures, etc. to support your claims and observations when it's proper (e.g. displaying examples, results, or system outputs).

Length: 4 pages (or more if necessary). Single space if MS word is used. Or you can choose latex templates, e.g. https://www.acm.org/publications/proceedings-template.

Grading: Based on each section described above, 25 points per section. As you can tell, they're related to each other. Each group member will make a separate submission with all group members' names indicated.

Sample papers for ideas

Search

Solving the Rubik's cube with deep reinforcement learning and search
https://www.nature.com/articles/s42256-019-0070-z
Task: solving rubik's cube using weighted A* and deep learning
Environment: rubik's cube
Evaluation metrics: solution length/percentage of instances being solved

Memory-Augmented Monte Carlo Tree Search
https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17139/15841
Task: implement a new design of MCTS that estimates a node value by augmenting the statistics of other node values
Environments: the game of Go
Baseline: Go program Fuego
Evaluation metrics: winning rate

Bidirectional Search That Is Guaranteed to Meet in the Middle
https://ojs.aaai.org/index.php/AAAI/article/view/10436
Task: implement a bidirectional search with a new heuristic function design
Environments: 10-pancake puzzle, rubik's cube
Baselines: iterative deepend A*
Evaluation metrics: number of node expansions

A+IDA*: A Simple Hybrid Search Algorithm*
https://www.ijcai.org/proceedings/2019/0168.pdf
Task: implement an algorithm that combines A* and IDA*
Environments: rubik's cube and 27/29-puzzles
Baselines: IDA*
Evaluation metrics: number of node expansions

Policy-Guided Heuristic Search with Guarantees
https://arxiv.org/abs/2103.11505

Monte Carlo Tree Search in Continuous Action Spaces with Execution Uncertainty
https://www.ijcai.org/Proceedings/16/Papers/104.pdf

*Monte Carlo *-Minimax Search*
https://www.ijcai.org/Proceedings/13/Papers/093.pdf

Multi-player Games

Monte Carlo Tree Search Techniques in the Game of Kriegspiel
https://www.ijcai.org/Proceedings/09/Papers/086.pdf
Task: solving Kriegspiel using MCTS
Environment: Kriegspiel
Evaluation metrics: elo rating

Searching Solitaire in Real Time
http://web.engr.oregonstate.edu/~afern/papers/solitaire.pdf
Task: solving Solitaire using real time search
Environments: Solitaire
Evaluation metrics: average time/%wins

Multiple Tree for Partially Observable Monte-Carlo Tree Search
Task: solving Phantom tic-toe-tok using MCTS
Environments: Phantom tic-toe-tok
Evaluation metrics: average time/%wins

DeepStack: Expert-level artificial intelligence in heads-up no-limit poker
https://www.science.org/doi/full/10.1126/science.aam6960
Task: solving heads-up no-limit poker using deep learning + CFR
Environment: poker
Evaluation metrics: exploitability

Deep Counterfactual Regret Minimization
https://arxiv.org/abs/1811.00164
Task: solving large-scale imperfect information game using CFR with deep neural networks as heuristic functions
Environments: poker
Evaluation metrics: exploitability

Bayesian Action Decoder for Deep Multi-Agent Reinforcement Learning
https://arxiv.org/pdf/1811.01458.pdf
Task: solve Hanabi using deep RL + public-belief formulation
Environment: Hanabi
Evaluation metrics: scores

Deep Reinforcement Learning from Self-Play in Imperfect-Information Games

https://arxiv.org/abs/1603.01121

RL

Rainbow: Combining Improvements in Deep Reinforcement Learning
https://arxiv.org/pdf/1710.02298.pdf
Task: doing an ablation study on different variants of deep Q-learning
Environments: Atari 2600 games
Baselines: different variants of DQN
Evaluation metrics: accumulated reward

Deep Reinforcement Learning that Matters
https://arxiv.org/pdf/1709.06560.pdf
Task: investigating the effect of hyperparameters/experimental setups for different deep RL methods
Environments: Atari games
Evaluation metrics: accumulated reward

Deep learning

Neural Architecture Search with Reinforcement Learning
https://arxiv.org/abs/1611.01578
Task: using RL to search neural architectures
Dataset: CIFAR
Model: RNN model description+REINFORCE training+layer skipption
Evaluation metrics: model accuracy

Semi-Supervised Classification with Graph Convolutional Networks
https://arxiv.org/abs/1609.02907

PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space
https://arxiv.org/abs/1706.02413

Natural Language Processing

A Study of Reinforcement Learning for Neural Machine Translation
https://aclanthology.org/D18-1397/
Task: using RL methods to train neural machine translation
Model: using MLE scores as RL rewards for NMT
Dataset: WMT14 EnglishGerman (En-De), WMT17 English-Chinese (EnZh) and WMT17 Chinese-English (Zh-En) translation dataset
Baseline: Transformer
Evaluation metrics: BLEU scores

A Deep Reinforced Model for Abstractive Summarization
https://arxiv.org/abs/1705.04304

Deep Reinforcement Learning for Dialogue Generation
https://aclanthology.org/D16-1127/

Bidirectional Attention Flow for Machine Comprehension
https://arxiv.org/pdf/1611.01603

Learning to Ask: Neural Question Generation for Reading Comprehension
https://aclanthology.org/P17-1123/