Interview Protocol

# *Consent form:* [*https://purdue.ca1.qualtrics.com/jfe/form/SV\_6Xw3lmOuVAG9y2a*](https://purdue.ca1.qualtrics.com/jfe/form/SV_6Xw3lmOuVAG9y2a)

*(Some demographic questions are included in the consent form.)*

# Research Questions

* + RQ1: What are the challenges and practices for PTNN selection?
  + RQ2: What attributes can be added to improve the model registries?
  + RQ3: Do the PTNNs available in model zoos accurately describe their behavior?

# Interview Questions

This interview is related to the trustworthy re-use of ***Pre-trained Neural Networks (*PTNNs*)***. Questions fall into 5 categories:

Background, Model Selection, Attributes, and Trustworthiness.

# 

# Demographic Questions (~5 min)

First, I would like to ask some questions regarding your background. The background questions can be answered briefly so that we can save time for the other interesting questions.

1. Could you briefly describe your role in your team?
2. Can you tell me a recent time when you used a pre-trained model from an external model hub?
   1. What challenge did you meet?
   2. How did you address it?

# Model Selection (~10 min)

The next set of questions is related to how you select a pre-trained neural network (PTNN). We are trying to understand the process that software engineers follow as they decide which PTNN to reuse in their projects.

1. Think about last time when you chose a PTNN from a model hub. How did you choose it? Can you summarize your decision-making process?
   1. Where do you select the PTNNs,
      1. pretrained models from a model hub or models serialized in github?
      2. Which hub do you choose to use?
      3. Why?
   2. When selecting the model, do you care more about the model’s performance than the architecture?
2. Do you think the PTNNs available in model registries accurately describe their behavior?
3. To what extent do the **discrepancies of performance metrics** *(i.e. the actual performances are different from the claimed performances)* affect your decision-making?
4. To what extent does the **robustness** of the models affect your decision?
5. To what extent does the **explainability** of the models affect your decision?
6. How frequently do you retrain the models (e.g. for PTNN validation or model adaptation/transfer learning)?
   1. Do you think lack of trainability is a problem when reusing a PTNN?
7. When selecting a PTNN from model registries, are there any other factors you consider?
8. What other challenges do you face when selecting a PTNN implementation from a model registry?

# Attributes (~10 min)

This set of questions relates to DL software attributes. We want to learn about what sort of information is useful to engineers who reuse PTNNs.

***(Show the slides)***

|  |  |  |
| --- | --- | --- |
| NPM defines the following attributes for Javascript Packages: | | |
|  |  |  |

1. What do you think would best help your team select a pre-trained model from model registries?

We define the following attributes:

## **Provenance**

*A measure of model lineage or traceability.*

1. Tell me about a time when you met **Provenance** problems when using PTNNs before (e.g. link to the paper, research prototype, GitHub page, custom website)?
   1. \*How did you address the problems?
2. Do you have other specific **Provenance** problems?
3. What would have been useful to know beforehand in order to solve those problems?

## **Reproducibility**

*The ability of a DL practitioner to produce the same accuracy and training/evaluation time from a PTNN as defined in its paper, source code, or group.*

1. Tell me about a time when you met any **Reproducibility** problems when using PTNNs before (e.g. runnability, configuration, data type, accuracy variance)?
   1. \*How did you address the problems?
2. Do you have specific **Reproducibility** problems?
3. What would have been useful to know beforehand in order to solve those problems?

## **Deployment Constraints**

*The ease with which an engineer can take a PTNN and reuse it in another environment, software project, etc.*

1. Tell me about a time when you met any **Deployment Constraint** problems when using PTNNs before?
2. Do you have specific deployment constraints? For example, latency, fine-tunability, energy consumption, accelerator
   1. \*How did you address the problems?
3. What would have been useful to know beforehand in order to solve those problems?

**Others**

Except for these three attributes, do you think there are any other attributes that would be helpful for PTNNs? (e.g. robustness, explianability)

# Trustworthiness (~5 min)

This set of questions is about PTNN trustworthiness. We are trying to understand how PTNN shortcomings affect engineers’ ability to rely on and reuse them.

1. Which aspects of the PTNN do you assume are trustworthy (e.g. the model architecture matches the paper; the performance with the weights is correct, ...)?
2. Have you found any discrepancies between the claimed pre-trained models and the downloaded version (e.g., accuracy, latency, architecture)?
   1. How did you find them?
   2. How did you address this problem?
   3. To what extent do you think these discrepancies are acceptable? (e.g. less than a certain percentage discrepancy of accuracy)
   4. What would you do if you find a model with discrepancies?
3. Do you think the discrepancies will have significant impacts? (For example, if the model is used in safety-critical systems.)

# 

# *Anonymization*

*Audio from the interviews will be recorded and transcribed by a trusted 3rd party service. The transcriptions will be viewed, and personal information will be anonymized. The audio recordings will be deleted after the transcripts are received, reviewed, and anonymized. A key mapping participant to the transcript will be kept separately by the research team until the conclusion of the study – this key may inform further participant selection, e.g. further probing of an interesting organization.*