A STUDY ON THE INTEGRATION OF PRE-TRAINED SSL, ASR, LM AND SLU MODELS FOR SPOKEN LANGUAGE UNDERSTANDING

Yifan Peng, Siddhant Arora*, Yosuke Higuchi, Yushi Ueda, Sujay Kumar, Karthik Ganesan, Siddharth Dalmia, Xuankai Chang, Shinji Watanabe*

Carnegie Mellon University, Pittsburgh, PA, USA

{yifanpen,siddhana,yhiguchi,yueda,sujayk,karthikg,sdalmia,xuankaic,swatanab}@andrew.cmu.edu

ABSTRACT

Collecting sufficient labeled data for spoken language understanding (SLU) is expensive and time-consuming. Recent studies achieved promising results by using pre-trained models in low-resource scenarios. Inspired by this, we aim to ask: which (if any) pre-training strategies can improve performance across SLU benchmarks? To answer this question, we employ four types of pre-trained models and their combinations for SLU. We leverage *self-supervised* speech and language models (LM) pre-trained on large quantities of unpaired data to extract strong speech and text representations. We also explore using *supervised* models pre-trained on larger external automatic speech recognition (ASR) or SLU corpora. We conduct extensive experiments on the SLU Evaluation (SLUE) benchmark and observe *self-supervised* pre-trained models to be more powerful, with pre-trained LM and speech models being most beneficial for the Sentiment Analysis and Named Entity Recognition task, respectively. [1]

Index Terms— spoken language understanding, low resource, pre-trained models

1. INTRODUCTION

Spoken language understanding (SLU) aims to extract semantics from a spoken utterance, which is essential for spoken dialog systems, voice assistants and intelligent home devices [1, 2]. SLU comprises a wide range of tasks, including extracting the intent [3, 4, 5] and associated entities [4, 6], recognizing emotion [7] for a given utterance, or modeling the topic of user conversations [8, 9]. Traditional SLU systems consist of two cascaded modules, i.e., automatic speech recognition (ASR) and natural language understanding (NLU). Recent studies have explored deep learning-based end-to-end (E2E) approaches that directly predict semantic meanings from a speech signal without converting it to intermediate text [10, 11, 12]. These E2E approaches avoid the error propagation seen in pipeline models as well as can capture non-phonemic signals such as pauses and intonations that a text-based system cannot capture.

However, E2E models usually require a large amount of labeled training data. SLU datasets are often expensive and time-consuming to collect, and hence most publicly available SLU datasets are limited in size. For low-resource applications, researchers have explored pre-trained representations and achieved promising results [13, 14]. Pre-trained language models like BERT [15] and DeBERTa [16] learn rich textual representations from unlabeled

---

*Equal contribution.

[1] Our code and models will be publicly available as part of the ESPnet-SLU toolkit.

text and are shown to advance the state-of-the-art (SOTA) performance when fine-tuned on downstream NLU tasks. Similarly, self-supervised speech representations can improve various speech processing tasks [13, 17]. Inspired by these studies, there has been a lot of interest in pre-training the acoustic [18, 12, 19] and semantic [20, 21, 22, 18, 19] model components for SLU tasks on large quantities of unlabeled speech and text data.

To this end, we ask the following questions: (i) Can pre-training methodologies help to advance performance across various SLU benchmarks? (ii) Which pre-training methodologies are most useful to improve performance for a given SLU task? (iii) Can we identify the kind of spoken utterances that are responsible for the majority of performance gains achieved by a given pre-training strategy? We seek to answer these questions by conducting a thorough study of various pre-training paradigms in the context of SLU. We investigate the following four types of pre-trained models and their combinations: 1) *self-supervised* learning (SSL) speech models [23, 24, 17, 25, 26], to generate powerful acoustic representations from the raw audio; 2) *self-supervised* language models (LM) [15, 16], to build strong semantic representations; 3) *supervised* ASR models pre-trained on large corpora [27, 28] and 4) *supervised* SLU models pre-trained on other SLU corpora [7, 29]. We conduct extensive experiments on the newly released Spoken Language Understanding Evaluation (SLUE) benchmark [14], which provides well-designed datasets with baselines and metrics for evaluating low-resource SLU. It consists of two SLU tasks: sentiment analysis (SA) on SLUE-VoxCeleb and named entity recognition (NER) on SLUE-VoxPopuli, with a small amount of labeled data to fine-tune the SLU system. This makes it an interesting benchmark to evaluate the efficacy of different pre-training approaches since the SLU dataset is particularly under-resourced.

Our contributions are as follows:

- We investigate the efficacy of four types of pre-trained models and their integrations in the context of SLU.

- We conduct extensive experiments on the SLUE benchmark and show that each pre-training approach can improve performance over the baseline E2E model without pre-training. Our best models can outperform the baseline by a large margin on both SA and NER tasks.

- Our results demonstrate that pre-training methodologies based on *self-supervised* learning are more powerful than those based on *supervised* learning. We hypothesize that this is because *self-supervised* models are trained on huge amounts of unlabeled data, which have extensive coverage of acoustic and linguistic variations. We also observe strong semantic representations from a pre-trained LM DeBERTa to be most helpful for the SA task, whereas strong speech

representations produced by an SSL model WavLM to be most beneficial for the NER task. We believe that future research to build SLU systems should employ pre-training paradigms based on *self-supervised* representations to boost model performance, particularly in low resource scenarios.

- We analyze the performance gains from pre-training techniques and find that most of the performance improvement from *self-supervised* pre-training methods can be seen in semantically and acoustically challenging utterances.

- Another interesting finding from our experiments is that the word error rate (WER) in ASR transcripts is not very well correlated with the downstream SA task but is a good indicator of the downstream NER performance.

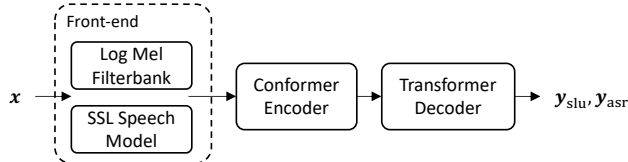## 2. METHODS

2.1. Problem formulation

As in ESPnet-SLU [30], we formulate the two SLU tasks (i.e., SA and NER) as a unified sequence-to-sequence problem. The input is a sequence of speech features extracted from the raw audio, and the output is a sequence of tokens consisting of the transcript and SLU labels. For SA, a sentiment label is prepended to the transcript. For NER, each entity phrase in the transcript begins with an entity tag and ends with a special token, which is consistent with the SLUE toolkit [14]. Figure 1 shows our SLU systems. The attention-based encoder-decoder architecture is adopted in our end-to-end (E2E) approaches. Specifically, we employ the Conformer [31] encoder and Transformer [32] decoder. Note that we do not use a language model for decoding. To better incorporate semantic information, we also exploit a two-pass approach [33], as introduced in Section 2.3.
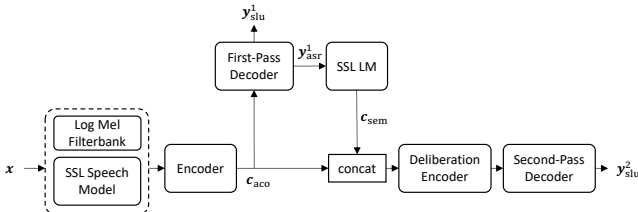
2.2. Self-supervised pre-trained speech models

Self-supervised speech models are pre-trained on large volumes of unlabeled speech data and can generate powerful representations for downstream tasks, especially for low-resource applications. We employ pre-trained speech representations to replace the commonly used log Mel filterbank features. Following prior work [13, 34], a weighted sum of multiple hidden states is utilized. During training, the parameters of pre-trained models are frozen and never updated. Five self-supervised speech models are evaluated, namely TERA [23], VQ-APC [24], Wav2Vec2 [17], HuBERT [25] and WavLM [26]. These models are trained using different objectives and corpora (see Table 1).

2.3. Self-supervised pre-trained language models

Self-supervised LMs are pre-trained on a large amount of unlabeled text data, which can generate high-quality semantic representations. There are multiple ways to utilize pre-trained LMs, such as pipeline approaches [14] and jointly modeling speech and text in a shared latent space [22]. Inspired by prior work on two-pass ASR [35, 36], we adopt a two-pass SLU approach [33] in this work, where the first pass predicts SLU labels and ASR hypotheses from the audio, and the second pass improves on the initial prediction by combining both acoustic and semantic information from ASR hypotheses. More specifically, the input speech (x) is first passed through an acoustic encoder to generate acoustic embeddings (c_{aco}). These embeddings are then passed to the first-pass decoder, which predicts the first-pass SLU labels (y^1_{slu}) and ASR transcript (y^1_{asr}).

(a) Our E2E SLU system. Self-supervised speech representations can replace the log Mel filterbank features. The entire model can be pre-trained on external ASR or SLU corpora.

(b) Our two-pass SLU system [33]. Self-supervised speech representations can replace the log Mel filterbank features. The 2nd pass attends to both acoustic information from 1st pass and semantic information from ASR transcript, as discussed in Section 2.3.

Fig. 1: Overview of our SLU systems.

Table 1: Summary of self-supervised pre-trained speech models used in this work. The Mix 94k dataset is a mixture of LibriLight 60k [37], GigaSpeech 10k [27], and VoxPopuli 24k [38].

Model	Architecture	Dataset	Objective
TERA	3-Trans	LibriSpeech 960h	masking
VQ-APC	3-GRU	LibriSpeech 960h	auto-regressive
Wav2Vec2	7-Conv 24-Trans	LibriLight 60k	contrastive
HuBERT	7-Conv 24-Trans	LibriLight 60k	pseudo-labeling
WavLM	7-Conv 24-Trans	Mix 94k	pseudo-labeling

The ASR transcript is then tokenized and processed by a pre-trained LM to generate semantic embeddings (c_{sem}). The acoustic and semantic embeddings are concatenated along the time dimension to form a joint embedding. Finally, the joint embedding is passed through a deliberation encoder before entering the second-pass decoder to predict more accurate second-pass SLU labels (y^2_{slu}). In this work, we have only experimented with BERT [15] and DeBERTa [16]; however, our method can incorporate any of the pre-trained models provided by HuggingFace. [2]

2.4. Supervised pre-trained ASR models

To compensate for the lack of labeled data in the under-resourced SLU task, we explore pre-training an SLU model using a large-scale external ASR corpus. This pre-training is expected to initialize the model with strong acoustic processing ability, which can improve performance on the downstream SLU task when fine-tuned on the target dataset.

In this work, we adopt GigaSpeech [27] and SPGISpeech [28] for the ASR pre-training, which are publicly available corpora consisting of 10k and 5k hours of transcribed English speech, respectively. We then initialize the SLU model with the pre-trained parameters, except for the embedding and softmax layers in the encoder and decoder networks. Then, the model is fine-tuned for a target task using a small amount of labeled data.

---

[2] https://huggingface.co/models

Table 2: Overview of the two datasets in the SLUE benchmark [14].

Dataset	Tasks	Size (utterances / hours)		
		Train	Dev	Test
SLUE-VoxCeleb	ASR, SA	5,777 / 12.8	955 / 2.1	4,052 / 9.0
SLUE-VoxPopuli	ASR, NER	5,000 / 14.5	1,753 / 5.0	1,842 / 4.9

2.5. Supervised pre-trained SLU models

When the target task has limited data, it is natural to pre-train the SLU model using other corpora designed for a similar task and then fine-tune it for the target task. For SA, we use existing emotion or sentiment datasets, IEMOCAP [7] and Switchboard (SWBD) Sentiment [29], which contain 12 and 140 hours of labeled speech data, respectively. The nine emotion labels in IEMOCAP can be optionally converted into three sentiment labels in the following manner: {happiness, excited, surprised: Positive}, {neutral: Neutral}, and {fear, sadness, anger, frustration, disgust: Negative}. The SWBD Sentiment dataset has three sentiment labels (Positive, Neutral, Negative). Thus, the labels in the two datasets can be preprocessed to match the three labels used in the SLUE SA task. For NER, we pre-train the model on the SLURP [4] dataset, which contains about 100 hours of audio data collected from single-turn user interactions with a home assistant. We use the original entity tags for pre-training.

2.6. Combination of pre-trained models

The pre-trained models can be combined at either model-level or output-level. For model-level combination, we employ SSL speech representations to replace the log Mel filterbank features. We then pre-train the entire encoder-decoder model on external corpora and fine-tune it using the target SLU dataset. Besides, we combine the self-supervised speech and language models in our two-pass approach (Section 2.3). For output-level combination, we adopt voting-based strategies to aggregate the decoded sequences from different models. We apply the majority voting to obtain SA results. As introduced in Section 2.1, the named entity tags are inserted into the transcript, so we directly apply the recognizer output voting error reduction (ROVER) [39] method to combine multiple hypotheses and extract NER results from the combined word sequence. We also obtain the ASR results for both tasks using ROVER.

3. EXPERIMENTAL SETUP

3.1. Datasets and tasks

We adopt the recently released SLUE benchmark [14] for evaluation, which focuses on naturally produced speech and contains limited labeled training data. The SLUE benchmark consists of two well-designed datasets, i.e., SLUE-VoxCeleb and SLUE-VoxPopuli, and three tasks, i.e., SA, NER and ASR. Specifically, SLUE-VoxCeleb is used for ASR and SA, while SLUE-VoxPopuli is used for ASR and NER. Details about each dataset are shown in Table 2. The training set is very limited, which is suitable for evaluating low-resource SLU. The released test sets are blind without groundtruth labels. We compare different methods using the development set.

3.2. Evaluation metrics

We adopt the evaluation metrics in the SLUE benchmark [14]. ASR is evaluated using word error rate (WER). SA aims to classify an input utterance as having negative, neutral, or positive sentiment,

which is evaluated using macro-averaged (unweighted) recall and F1 scores. Since the negative class has only 3 instances in the official development set, we find the results to be unstable. [3] Hence, we use the macro F1 (F1*) and recall (Recall*) computed only on positive and neutral classes for model comparison. [4] NER focuses on recognizing the named entities and their tags (types) in a given utterance. There are 7 distinct entity tags. We use micro-averaged F1 and label-F1 scores for NER. The F1 score considers an unordered list of named entity phrases and tag pairs, while the label-F1 only considers the tag predictions and ignores the potential misspelling and segmentation errors in speech recognition.

3.3. Implementation Details

Our models are implemented in PyTorch [40], and the experiments are conducted using the ESPnet-SLU toolkit [30, 41]. The self-supervised pre-trained speech and language models are obtained from S3PRL [13], Fairseq [42] and HuggingFace [43], while the pre-trained ASR and SLU models are downloaded from ESPnet Model Zoo. Our models are based on the attention-based encoder-decoder architecture described in Section 2.1. The encoder is a 12-layer Conformer [31], while the decoder is a 6-layer Transformer [32]. The number of heads and dimension of a self-attention layer are set to 4 and 256, respectively. The linear units are 1024 for the encoder and 2048 for the decoder. During training, speed perturbation and SpecAugment [44] are performed for data augmentation. We apply dropout [45] and label smoothing [46] to mitigate overfitting. We use the Adam [47] optimizer with a maximum learning rate of 2e-3 and a weight decay of 1e-6 or 1e-5. We also employ the Transformer learning rate scheduler [32] with 5k warmup steps. We perform joint CTC-attention training and decoding [48, 49]. More details about our models and the config files will be publicly available as part of the ESPnet-SLU [30] toolkit.

4. RESULTS

Table 3 presents the SA results on SLUE-VoxCeleb, and Table 5 shows the NER results on SLUE-VoxPopuli. In the following subsections, we discuss the effect of different pre-trained models on the performance of these SLU tasks.

4.1. Sentiment Analysis

Table 3 shows that all pre-trained approaches boost SA performance on the SLUE-VoxCeleb dataset. Among all the models with SSL features, the model using WavLM features achieves the best recall (Recall*: 66.9) and F1 (F1*: 66.9). Further, models with HuBERT and Wav2Vec2 also outperform the baseline, showing that self-supervised speech representations are more powerful than the log Mel features in the low-resource scenario.

For models pre-trained on large external ASR data, pre-training on GigaSpeech (Recall*: 66.3, F1*: 66.6) achieves better scores than pre-training on SPGISpeech, because GigaSpeech has larger and more diverse training data. However, the best *supervised* model

---

[3] A small change in the negative class will drastically affect the overall metric. For example, if a model achieves a recall of 60% in both positive and neutral classes but fails to recognize any negative instances, the macro-averaged recall will be 40%. If it happens to recognize one of the three negative instances, the recall of the negative class will be 33% and the macro-recall will be 51%. This means the metric averaged over all three classes is very unstable.

[4] After detailed discussions with the authors of the SLUE benchmark.

Table 3: Macro-averaged recall (Recall*) and F1 (F1*) scores (%) computed only on positive and neutral classes for sentiment analysis on SLUE-VoxCeleb. As discussed in section 3.2, we use Recall* and F1* for model comparison. LM decoding is not used. Bold values indicate the best performance obtained both with and without output-level system combinations on this dataset and X̲ indicates outperforming the SA performance of no pre-train model.

	Pre-trained Model/Corpus	Recall* (↑)	F1* (↑)	WER (↓)
Our E2E approaches				
w/o pre-train	N/A	62.4	63.6	33.0
	TERA	62.5̲	62.4	27.1
	VQ-APC	61.3	62.1	29.8
w/ SSL	Wav2Vec2	64.5̲	64.4̲	14.2
	HuBERT	64.5̲	65.2̲	12.8
	WavLM	**66.9**	66.9̲	9.1
w/ ASR	GigaSpeech	66.3̲	66.6̲	11.3
	SPGISpeech	63.3̲	64.1̲	14.2
w/ SLU	IEMOCAP	62.4	62.9	33.2
	SWBD Sentiment	64.7̲	64.8̲	21.9
	WavLM+IEMOCAP	64.3̲	65.2̲	9.3
	WavLM+SWBD Sentiment	64.9̲	65.7̲	9.0
w/ LM (2-pass)	BERT	64.7̲	65.1̲	29.5
	DeBERTa	66.2̲	**67.3̲**	29.5
	WavLM+BERT	66.8̲	65.7̲	9.4
	WavLM+DeBERTa	**66.9̲**	66.5̲	9.3
Our system combination				
majority voting	best 3 models	**67.9**	**69.0**	9.7

Table 4: Macro-averaged recall and F1 scores (%) for sentiment analysis on SLUE-VoxCeleb. As discussed in Section 3.2, we actually do not use these Recall and F1 for model comparison since they are unstable, but we do show that we outperform the results reported in the SLUE benchmark using a similar Wav2Vec2 based SLU model.

	Pre-trained Model	Recall (↑)	F1 (↑)	WER (↓)
SLUE benchmark [14]				
Oracle Text	BERT	43.0	43.6	0.0
	DeBERTa	55.6	46.5	0.0
Pipeline w/o LM	Wav2Vec2+DeBERTa	54.2	45.3	11.0
Pipeline w/ LM	Wav2Vec2+DeBERTa	55.1	45.8	9.1
E2E	Wav2Vec2 w/o LM	45.0	44.2	11.0
E2E	Wav2Vec2 w/ LM	45.0	44.2	9.1
Our E2E approach				
w/ SSL	Wav2Vec2	54.1	46.4	14.2

pre-trained on external ASR data has lower performance than the best model using *self-supervised* speech representations.

For models pre-trained on external sentiment datasets, the performance is higher when using the SWDB dataset, possibly due to the larger size of SWDB sentiment compared to IEMOCAP. Two-pass SLU models (Section 2.3) that use LMs to enhance semantic processing power perform better than those pre-trained on external SLU datasets. By incorporating DeBERTa as a pre-trained LM (F1*:67.3), we can outperform, in terms of F1 value, all other pre-training paradigms for SA.

Our results demonstrate that *self-supervised* speech models and LMs can generate more robust speech and semantic representations, respectively, underscoring the efficacy of leveraging *self-supervised* models to pre-train SLU systems. Another remarkable observation is that the WER of the two-pass SLU model with DeBERTa (WER: 29.5, F1*: 67.3) is much higher than that of the model using a WavLM frontend (WER: 9.1, F1*: 66.9) but the two-pass approach still achieves better SA performance. This result shows that WER

Table 5: Micro-averaged F1 and label-F1 scores (%) for named entity recognition on SLUE-VoxPopuli. LM decoding is not used in our approaches. Bold values indicate the best performance obtained both with and without output-level system combinations on this dataset and X̲ indicates outperforming the NER performance of no pre-train model.

	Pre-trained Model/Corpus	Label-F1 (↑)	F1 (↑)	WER (↓)
SLUE benchmark [14]				
Oracle Text	BERT	90.9	86.2	0.0
	DeBERTa	91.1	87.5	0.0
Pipeline w/o LM	Wav2Vec2+DeBERTa	83.5	63.3	14.0
Pipeline w/ LM	Wav2Vec2+DeBERTa	87.4	74.9	9.1
E2E w/o LM	Wav2Vec2	69.1	55.6	14.0
E2E w/ LM	Wav2Vec2	79.0	70.2	9.1
Our E2E approaches				
w/o pre-train	N/A	67.6	54.7	34.2
	TERA	70.9̲	57.1̲	28.6
	VQ-APC	76.6̲	63.6̲	24.1
w/ SSL	Wav2Vec2	83.3̲	69.5̲	12.8
	HuBERT	84.8̲	69.7̲	12.5
	WavLM	**88.0̲**	74.5̲	9.3
w/ ASR	GigaSpeech	86.0̲	73.9̲	11.2
	SPGISpeech	84.1̲	71.4̲	12.2
w/ SLU	SLURP	71.5̲	59.7̲	33.7
	WavLM+SLURP	87.5̲	**75.7̲**	9.0
w/ LM (2-pass)	BERT	69.2̲	54.5	33.8
	DeBERTa	69.4̲	55.5̲	34.1
	WavLM+BERT	87.3̲	73.5̲	9.6
	WavLM+DeBERTa	87.7̲	74.0̲	9.5
Our system combination				
ROVER	best 4 models	**88.7**	**77.2**	8.6

in ASR transcripts is not a good indicator of the downstream SA performance.

To compare with the E2E results from the SLUE [14] benchmark, we also report the original Recall and F1 values in Table 4. Our approach shows higher performance using the same Wav2Vec2 model (54.1 vs. 45.0 in recall, 46.4 vs. 44.2 in F1), demonstrating the effectiveness of our encoder-decoder-based SLU modeling.

We further investigate integrating speech and text pre-training approaches at the model-level using WavLM features as input for our two-pass SLU model (see WavLM+BERT, WavLM+DeBERTa in Table 3) as well as pre-training on external SLU datasets (see WavLM+IEMOCAP, WavLM+SWDB Sentiment in Table 3). We generally observe an improvement in performance in comparison to the model that uses log Mel Filterbank features, except for the two-pass SLU model with DeBERTa. We further experiment with output-level combinations (see Section 2.6) of our best three models, i.e., pre-trained with WavLM, GigaSpeech, and DeBERTa. As shown in Table 3 (see "Our system combinations"), this output-level model combination (Recall*: 67.9, F1*: 69.0) outperforms all the individual models, indicating that we can advance the performance on SLU benchmarks by combining different pre-training approaches.

4.2. Named Entity Recognition

Table 5 shows that similar to SA, all pre-training approaches improve NER performance over the baseline SLU model without pre-training. WavLM (F1: 74.5, Label F1: 88.0) performs the best in all SSL models, and our findings for the utility of different pre-trained SSL systems as feature extractors are mainly consistent with the SA task. We also observe that pre-training on external ASR data, par-

ticularly the GigaSpeech dataset (F1: 73.9, Label F1: 86.0), boosts performance but is still worse than using WavLM features.

We observe that models incorporating strong speech representations from the SSL model or external ASR dataset generally have better performance than those using semantic representations obtained through pre-trained LM or external SLU corpora. Using the *self-supervised* WavLM model to extract speech features is found to be most beneficial for advancing the NER performance of SLU systems. We also find the NER performance of all encoder-decoder models to be well correlated with the WER of ASR transcripts.

We compare our models with the results from the SLUE [14] benchmark and observe that even without LM decoding, our best model using WavLM features outperforms the SLUE E2E approach with LM decoding (88.0 vs. 79.0 in label-F1, 74.5 vs. 70.2 in F1).

We also experiment with a model that takes WavLM features as input and pre-trains using external LMs or SLU corpora (see WavLM+BERT, WavLM+DeBERTa, WavLM+SLURP in Table 5) and conclude that using WavLM features inside this pre-training framework can boost the NER performance. Further, by pre-training on the SLURP dataset, the model that uses WavLM features can achieve an F1 score of 75.7, which is higher than the model pre-trained using only WavLM features (F1: 74.5). This result shows that gains achieved by pre-training on SLURP are complementary to strong speech representations obtained from WavLM and provides evidence that integration of different pre-training paradigms at the model-level can advance the NER performance. We then investigate the output-level combination (see Section 2.6) of our best four models, i.e., pre-trained with WavLM, model-level combination of WavLM and SLURP, model-level combination of WavLM and DeBERTa and GigaSpeech. Results show that integrating different pre-training approaches using ROVER (see "Our system combinations" in Table 5) can further boost NER performance, which encourages future research on combinations of different pre-training approaches.

5. ANALYSIS

Recently, there has been interest [50] in quantifying the semantic complexity of SLU datasets and reporting the performance of a given SLU system across different semantic complexities. Prior work [51] has also shown that ASR pre-training can be particularly useful for acoustically and semantically challenging utterances. Inspired by these findings, we also compare the performance of our trained models across utterances of different acoustic and semantic complexities. To facilitate this analysis, we divide the development set into classes of different difficulties with roughly similar number of utterances. Our analysis helps us develop a deep understanding of gains achieved by best performing pre-trained models on each of the SLU tasks, i.e., using *self-supervised* DeBERTa for the SA task and *self-supervised* WavLM for the NER task. We further compare performance between *supervised* and *self-supervised* pre-training methodologies and characterize the utterances responsible for the performance gap between the two approaches.

5.1. Acoustic Analysis

Figure 2 analyzes the gains in NER performance by using SSL features as input. We quantify the acoustic complexity of spoken utterance using WER of ASR transcripts produced by our baseline E2E SLU model. We observe that the performance gap between the baseline model and the pre-trained models increases as the ASR difficulty of utterances increases in the VoxPopuli dataset. Further, when we compare using *self-supervised* WavLM features and *supervised* pre-training on GigaSpeech, most of the performance difference is ob-
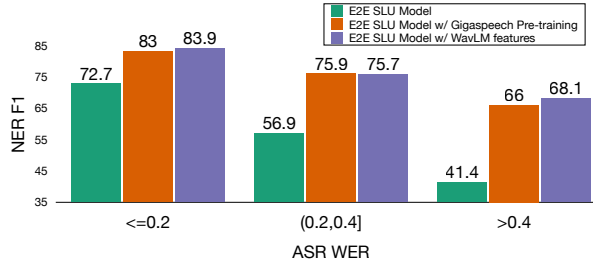

Fig. 2: Results comparing the NER performance of our models without pre-training, with ASR pre-training on GigaSpeech and with WavLM features as input across different ASR difficulties measured by WER of the no pre-train model on SLUE-VoxPopuli. The performance gain from using *self-supervised* WavLM increases as the ASR difficulty increases.
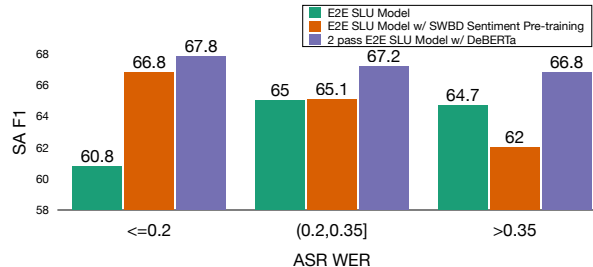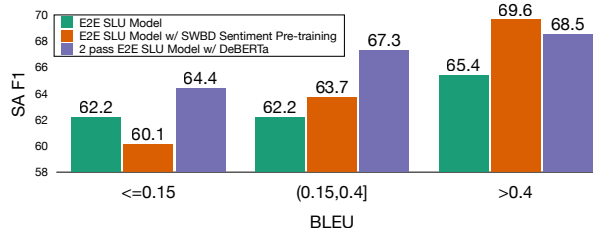

Fig. 3: Results comparing the SA performance of our models without pre-training, with SLU pre-training on SWDB Sentiment and with DeBERTa as a pre-trained LM across different ASR difficulties measured using WER of the no pre-train model on SLUE-VoxCeleb. The largest performance gap between the model using *self-supervised* DeBERTA and the model pre-trained on SWDB Sentiment is from acoustically challenging utterances (WER> 0.35).

served in extremely difficult utterances (i.e., WER > 0.4). Hence, we infer that SSL features are particularly beneficial for acoustically challenging utterances.
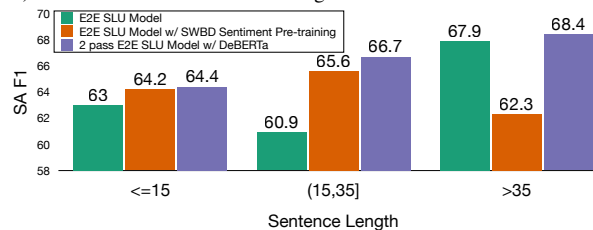
Prior work [33] has shown that better semantic modeling can help recover from ASR errors and hence can improve performance on acoustically challenging utterances. Inspired by this, we similarly analyze the performance gains of the two-pass SLU model with DeBERTa across different acoustic complexities in Figure 3. We do not observe any clear trend between SA performance and ASR WER for our models on the VoxCeleb dataset, which is consistent with our findings in Section 4.1. Interestingly, we observe that acoustically difficult utterances (WER > 0.35) account for the maximum performance gap between models pre-trained with *self-supervised* DeBERTa and *supervised* SWBD Sentiment SLU model, demonstrating that LM features are more robust to errors in ASR transcript.

5.2. Semantic Analysis

Figure 4 analyzes the performance of the two-pass SLU model that uses DeBERTa as a pre-trained LM. A spoken utterance with many unique n-grams not seen in training utterances makes the semantic understanding task more challenging [51]. As a result, we categorize the test utterances based on their n-gram overlap with training utterances. We choose the Sentence BLEU [52] score as a proxy for n-gram overlap and compute BLEU-4 to quantify the semantic complexity of a given utterance. Figure 4a shows that SA performance generally seems to improve for all models as lexical overlap with training utterance increases. The two-pass model is observed to

(a) SA performance across different lexical overlap (measured by BLEU score) of test utterances with the training set.
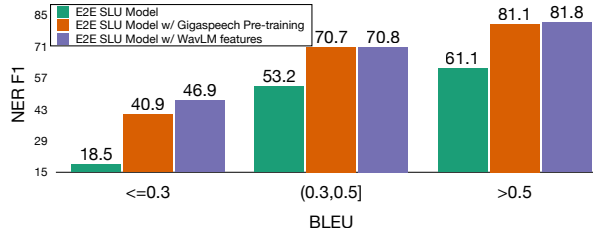

(b) SA performance across length of test utterances.

Fig. 4: Results comparing the SA performance of models without pre-training, with SLU pre-training on SWDB Sentiment and with DeBERTa as a pre-trained LM across different semantic difficulties on SLUE-VoxCeleb. The semantic difficulty is measured using (a) lexical overlap with training utterances and (b) utterance length. We observe that *self-supervised* DeBERTa representations are particularly useful for semantically complex utterances which have low n-gram overlap with the training set or longer utterance length.
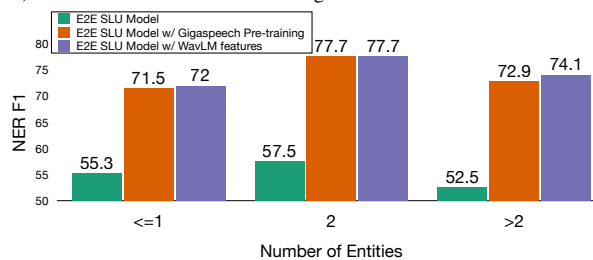
be better than the baseline SLU model in all categories of semantic difficulty. Further, the performance gap between the baseline SLU model and the two-pass SLU model with DeBERTa is greatest for utterances with BLEU scores in the bucket of (0.15,0.4]. Both models seem to be similarly struggling on more challenging utterances (i.e., for BLEU<= 0.15). Compared with the model pre-trained on the SWBD sentiment dataset, the two-pass SLU system seems particularly helpful for semantically more complex utterances.

Another way to quantify semantic difficulty is using the length of the ASR transcript for a given utterance. Figure 4b shows no clear trend between SA performance and transcript length for all models. However, we observe that the performance gap between the model pre-trained on the SWBD sentiment dataset and the two-pass SLU model with DeBERTa increases with an increase in transcript length. We conclude that most of the performance difference between *self-supervised* LM and *supervised* model trained on an external SLU dataset is for semantically challenging utterances which have low n-gram overlap with training utterances or longer utterance length.

We perform a similar analysis for the NER model using WavLM features as shown in Figure 5. Figure 5a shows that NER performance for all models improves with an increase in lexical overlap with training utterances. The performance gains achieved by both ASR pre-training and SSL features are highest for challenging utterances (i.e., BLEU <= 0.3). Remarkably, most of the performance gap between WavLM and GigaSpeech model is also on utterances with high semantic complexity, probably due to extensive linguistic and acoustic variations in large amounts of unlabelled data used to train SSL speech models. Figure 5b breakdowns NER performance based on the number of entities in a test utterance. An utterance with many entity mentions requires better semantic understanding. Again, we observe that *self-supervised* speech representations are more robust to semantic complexity (Entity No. > 2) than *super-*

(a) NER performance across different lexical overlap (measured by BLEU score) of test utterances with the training set.

(b) NER performance across different number of entities in test utterances.

Fig. 5: Results comparing the NER performance of models without pre-training, with ASR pre-training on GigaSpeech and with WavLM features as input across different semantic difficulties on SLUE-VoxPopuli. The semantic difficulty is measured using (a) lexical overlap with training utterances and (b) number of entities. We observe that *self-supervised* WavLM representations are particularly useful for semantically complex utterances which have low n-gram overlap with the training set or many entity mentions.

vised representations pre-trained on an ASR dataset.

6. CONCLUSION

In this work, we present a thorough analysis of four types of pre-training approaches for SLU. We show that each of the pre-trained models can boost performance over the baseline SLU model without pre-training. Our results show that *self-supervised* pre-trained models achieve higher performance than *supervised* pre-trained models. Specifically, we demonstrate that SSL speech models give the most performance gains for the NER task and pre-trained LMs give the greatest performance improvement on the SA task. We also observe that gains achieved by different pre-training methodologies are complementary to each other and by combining different approaches, we can further advance the SLU performance. Finally, we show a detailed analysis to gain insights into our performance gains and infer that *self-supervised* pre-trained models are particularly beneficial for acoustically and semantically challenging utterances.

We recommend future studies to leverage *self-supervised* representations to advance SLU performance, particularly for under-resourced settings. We hope insights derived from our study will facilitate future research on the tight integration of pre-training methodologies for SLU.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] Dian Yu, Michelle Cohn, Yi Mang Yang, Chun-Yen Chen, Weiming Wen, et al., "Gunrock: A social bot for complex and engaging long conversations," in *Proc. EMNLP-IJCNLP: System Demonstrations*, 2019.

[2] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al., "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[3] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, "Speech model pre-training for end-to-end spoken language understanding," in *Proc. Interspeech*, 2019.

[4] Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser, "SLURP: A spoken language understanding resource package," in *Proc. EMNLP*, 2020.

[5] Alaa Saade, Alice Coucke, Alexandre Caulier, Joseph Dureau, Adrien Ball, Théodore Bluche, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, et al., "Spoken language understanding on the edge," *arXiv preprint arXiv:1810.12735*, 2018.

[6] Miguel Del Rio, Natalie Delworth, Ryan Westerman, Michelle Huang, Nishchal Bhandari, Joseph Palakapilly, Quinten Mc-Namara, Joshua Dong, Piotr Żelasko, and Miguel Jetté, "Earnings-21: A practical benchmark for ASR in the wild," in *Proc. Interspeech*, 2021.

[7] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Changa, S. Lee, and S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Proc. LREC*, vol. 42, no. 4, pp. 335–359, 2008.

[8] Daniel Ortega and Ngoc Thang Vu, "Lexico-acoustic neural-based models for dialog act classification," in *Proc. ICASSP*, 2018.

[9] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, et al., "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics*, vol. 26, no. 3, pp. 339–371, 2000.

[10] Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *Proc. SLT*, 2018.

[11] Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio, "Towards end-to-end spoken language understanding," in *Proc. ICASSP*, 2018.

[12] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, "Speech model pre-training for end-to-end spoken language understanding," *arXiv preprint arXiv:1904.03670*, 2019.

[13] Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, et al., "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech*, 2021.

[14] Suwon Shon, Ankita Pasad, Felix Wu, Pablo Brusco, Yoav Artzi, Karen Livescu, and Kyu J Han, "SLUE: New benchmark tasks for spoken language understanding evaluation on natural speech," *arXiv preprint arXiv:2111.10367*, 2021.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL HLT*, 2019.

[16] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen, "DeBERTa: decoding-enhanced BERT with disentangled attention," in *Proc. ICLR*, 2021.

[17] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Proc. NeurIPS*, 2020.

[18] Cheng-I Lai, Yung-Sung Chuang, Hung-Yi Lee, Shang-Wen Li, and James Glass, "Semi-supervised spoken language understanding via self-supervised speech and language model pretraining," in *Proc. ICASSP*, 2021.

[19] Ankita Pasad, Felix Wu, Suwon Shon, Karen Livescu, and Kyu J Han, "On the use of external data for spoken named entity recognition," *arXiv preprint arXiv:2112.07648*, 2021.

[20] Yung-Sung Chuang, Chi-Liang Liu, Hung yi Lee, and Lin shan Lee, "SpeechBERT: An audio-and-text jointly learned language model for end-to-end spoken question answering," in *Proc. Interspeech*, 2021.

[21] Bhuvan Agrawal, Markus Müller, Martin Radfar, Samridhi Choudhary, Athanasios Mouchtaris, and Siegfried Kunzmann, "Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding," *arXiv preprint arXiv:2011.09044*, 2020.

[22] Yu-An Chung, Chenguang Zhu, and Michael Zeng, "SPLAT: Speech-language joint pre-training for spoken language understanding," *arXiv preprint arXiv:2010.02295*, 2020.

[23] Andy T Liu, Shang-Wen Li, and Hung-yi Lee, "TERA: Self-supervised learning of transformer encoder representation for speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2351–2366, 2021.

[24] Yu-An Chung, Hao Tang, and James Glass, "Vector-quantized autoregressive predictive coding," *arXiv preprint arXiv:2005.08392*, 2020.

[25] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451–3460, 2021.

[26] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al., "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *arXiv preprint arXiv:2110.13900*, 2021.

[27] Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al., "GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.

[28] Patrick K. O'Neill, Vitaly Lavrukhin, Somshubra Majumdar, Vahid Noroozi, Yuekai Zhang, et al., "SPGISpeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition," in *Proc. Interspeech*, 2021.

[29] Eric Chen, Zhiyun Lu, Hao Xu, Liangliang Cao, Yu Zhang, and James Fan, "A large scale speech sentiment corpus," in *Proc. LREC*, 2020.

[30] Siddhant Arora, Siddharth Dalmia, Pavel Denisov, Xuankai Chang, Yushi Ueda, Yifan Peng, Yuekai Zhang, Sujay Kumar, Karthik Ganesan, Brian Yan, et al., "ESPnet-SLU: Advancing spoken language understanding through ESPnet," *Proc. ICASSP*, 2022.

[31] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Proc. NeurIPS*, 2017.

[33] Siddhant Arora, Siddharth Dalmia, Xuankai Chang, Brian Yan, Alan Black, and Shinji Watanabe, "Two-pass low latency end-to-end spoken language understanding," in *Arxiv preprint arXiv:2207.06670*, 2022.

[34] Xuankai Chang, Takashi Maekaku, Pengcheng Guo, Jing Shi, Yen-Ju Lu, Aswin Shanmugam Subramanian, Tianzi Wang, Shu-wen Yang, Yu Tsao, Hung-yi Lee, et al., "An exploration of self-supervised pretrained representations for end-to-end speech recognition," *arXiv preprint arXiv:2110.04590*, 2021.

[35] Ke Hu, Tara N Sainath, Ruoming Pang, and Rohit Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *Proc. ICASSP*, 2020.

[36] Tara N. Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mirkó Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, Ian McGraw, and Chung-Cheng Chiu, "Two-pass end-to-end speech recognition," in *Proc. Interspeech*, 2019.

[37] Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al., "Libri-Light: A benchmark for ASR with limited or no supervision," in *Proc. ICASSP*, 2020.

[38] Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," *arXiv preprint arXiv:2101.00390*, 2021.

[39] Jonathan G Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proc. ASRU*, 1997.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "PyTorch: An imperative style, high-performance deep learning library," *Proc. NeurIPS*, 2019.

[41] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018.

[42] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[43] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush, "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP: System demonstrations*, 2020.

[44] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019.

[45] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[46] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton, "When does label smoothing help?," *Proc. NeurIPS*, 2019.

[47] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[48] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*, 2017.

[49] Takaaki Hori, Shinji Watanabe, and John R Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proc. ACL*, 2017.

[50] Joseph P. McKenna, Samridhi Choudhary, Michael Saxon, Grant P. Strimel, and Athanasios Mouchtaris, "Semantic complexity in end-to-end spoken language understanding," in *Proc. Interspeech*, 2020.

[51] Siddhant Arora, Alissa Ostapenko, Vijay Viswanathan, Siddharth Dalmia, Florian Metze, Shinji Watanabe, and Alan W. Black, "Rethinking end-to-end evaluation of decomposable tasks: A case study on spoken language understanding," in *Proc. Interspeech*, 2021.

[52] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. ACL*, 2002.

[53] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, "XSEDE: Accelerating scientific discovery," *Computing in Science & Engineering*, vol. 16, no. 5, pp. 62–74, 2014.

[54] Nicholas A Nystrom, Michael J Levine, Ralph Z Roskies, and J Ray Scott, "Bridges: a uniquely flexible HPC resource for new communities and data analytics," in *Proc. XSEDE*, 2015.