A ACRONYMS LIST

The acronyms concerning concepts and methods, and also in our framework, are shown in Table 8,9 respectively.

Table 8: Acronyms concerning concepts and methods

Acronyms	Meaning
IND	In-Domain
OOD	Out-of-Domain
QA systems	Question and Answering systems
BERT	Bidirectional Encoder Representations from Transformers
RoBERTa	Robustly optimized BERT approach
GAN	Generative Adversarial Network
Seq-GAN	Sequence GAN
DCGAN	Deep Convolutional GAN
GPT-2	Generative Pre-Training 2.0
GRU	Gated Recurrent Units
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional LSTM
MLP	Multi Layer Perceptron
Relu	Rectified Linear Units
KL-divergence	Kullback–Leibler divergence
AUROC	Area Under the Receiver Operating Characteristic curve
AUPR	Area Under Precision-Recall curve

Table 9: Acronyms In Our Framework

Acronyms	Meaning
IDC	In-Domain Classifier
ODC	Out-of-Doamin classifier
ENC	Encoder of autoencoder
DEC	Decoder of autoencoder
G	Generator of GAN
D	Discriminator of GAN

B DATASET INFORMATION

CLINC dataset: this intent classification dataset is available at https://github.com/clinc/oos-eval. It has 22,500 IND questions across 150 intents and also provides 1,200 OOD questions. Table 10 shows some examples from this dataset.

SNIPS dataset: this dataset is collected from the Snips voice platform, which is available at https://github.com/sonos/nlu-benchmark/tree/master/2017-06-custom-intent-engines. It contains 15,000 texts across 7 intents. According to our OOD category choice policy, we selected one 'play music' category as the OOD category due to its similar, but different presentation to another existing 'add to playlist' category. Table 11 shows some examples from this dataset.

FB-Multi dataset: this dataset is drawn from the Facebook multilingual database, which is available at available at https://fb.me/multilingual_task_oriented_data. We removed the 'weather find' category due to its extremely large size (over 10k records) which caused data imbalances, and the noise it contains due to international place names. The dataset contains over 30,000 inquires across 12 intents, but their distribution is very unbalanced. Given their similarity with the remaining 'set/cancel reminder' and 'modify/cancel alarm' categories, we selected the 'show reminder' and 'show alarm' categories as the manual OOD categories. Table 12 shows some examples from this dataset.

Table 10: Examples from CLINC dataset

Intent	Example
card_declined	can you help me understand why my card got declined
distance	what time will i get to the beach taking the bus
oil_change_how	can you teach me how to change my oil in my car
lost_luggage	where would i find my luggage
plug_type	what kind of electrical outlet does that country use
…	…
OOD(predefined)	tell me the steps as to how to begin a career as a journalist where can i find the best divorce lawyer

Table 11: Examples from SNIPS dataset

Intent	Example
AddToPlaylist	add nana tanimura to a sudden rainstorm i d like johnny nash to be put into my playlist always pop punk
BookRestaurant	book sot for 22 minutes from now at a restaurant with parking
GetWeather	what is the weather in sehlabathebenationalpark
RateBook	rate this current essay four out of 6
SearchCreative Work	find the song called international journal of bilingualism
SearchScreening Event	what films are playing at the nearest movie theatre
PlayMusic (as OOD cate.)	play playlist the realest down south i want to listen to the album going back to the blue ridge mountains on iheart

C TRAINING DETAILS

C.1 Implementation for Proposed Model

Our experiments were performed on a computer with Ubuntu 18.04 system, 128GB RAM, a Nvidia RTX2080 GPU and an Intel i9-10980XE CPU. The language/library used to write the neural network is Python3.7.6/Pytorch1.9.0 .

We used Adam stochastic optimization algorithm as the optimizer for all our networks. For autoencoder, the learning rate was set to

Table 12: Examples from FB-Multi dataset

Intent	Example
alarm/modify_alarm	Move my 8:45am today alarm to the next day please
alarm/cancel_alarm	Remove my alarm originally scheduled for 6 PM Sunday
alarm/set_alarm	set my alarm everyday but sunday at 8
alarm/snooze_alarm	Please rest the alarm for 4 PM
alarm/time_left_on_alarm	How much longer left on my 1PM alarm
reminder/set_reminder	can you send me a reminder to bring my coupons with me on Thursday
reminder/cancel_reminder	Cancel my reminder about my dentist appointment
weather/checkSunrise	When does the sun rise on Saturday
weather/checkSunset	When will the sun set in Delaware
alarm/show_alarms (as OOD cate.)	how many alarms do I have set for today before 12pm List all of my upcoming alarms for next week
reminder/show_reminders (as OOD cate.)	Did I already set a reminder for the trash tomorrow Is scheduling a doctor appointment already on my reminder list

3e-4, and we trained it about 30 to 35 epochs until the reconstruction loss was below 0.2 [11]. The reason is that as the decode target is not only IND but also fake-OOD, we found that under-trained or over-trained autoencoder will all have an negative impact for the subsequent GAN-OOD generation training. For the IND classifier, a learning rate of 1e-3 was used and 7 epochs' training was taken. For the GAN, we applied the learning rate of 2e-4 for both D and G, and trained 200 epochs.
Finally, for the OOD classifier (ODC), we trained it for 20 epochs with a learning rate of 2e-4. The training data was composed with the INDs and generated fake-OODs of same size to the INDs. We performed the training for 10 times and generated different fake-OODs each time. The results were averaged to be the final result.

C.2 Implementation for Baselines

For the *'traditional classifier output-based'* baseline methods, the same TextCNN and configurations were used as described in Appendix C.1.
For the *'large-scale network-based'* baseline methods, we used huggingface's *bert-base-uncased* pre-trained model [12]. Before applying baseline methods, it was trained (finetuned) with the texts and labels from IND data for 4 epochs. The optimizer is AdamW, and the learning rate is 2e-5.
For the other two adversarial learning methods ***En/De Recons.*** and ***GAN/DIS***, which utilized autoencoder and GAN, we use the same training configuration as in Appendix C.1, but the differences are they were trained only on the IND dataset, and the additional steps like 'secondary training for G' is not needed.

[11] The learning rate 1e-3 was also applicable and faster but we found it sometimes made autoencoder training could not converge
[12] https://huggingface.co/transformers/pretrained_models.html