# COMPSCI 682, Fall 24
# Project Spotlights

Day 2 – Dec 5, 2024

Umass, Amherst

# Instructions, once again

Speakers will have 2 mins to present their work

We will warn you at when 1 min, 30 sec, 0 sec remain

Must wrap up at 0

We will ask questions during grading

Thanks!

# But first attendance

# Presentation order

| | | |
|---|---|---|
| 1 | zero-shot knowledge graph embedding | Mingchen Li (mingchenli@umass.edu), Rohan Pandey (rohanpandey@umass.edu) |
| 9 | medical image anomaly detection using VLM | Deepti Guntur (dguntur@umass.edu), Lucy Bodtman (lbodtman@umass.edu) |
| 12 | domain adaptation for OCR models | Frankie Furnelli (ffurnelli@umass.edu), Thomas Potts (tpotts@umass.edu), Virtulya Rajput (vrajput@umass.edu) |
| 13 | Optimizing Few-Shot Learning: A Semi-supervised hybrid approach for enhanced classification on Omniglot | Rutika Bhoir (rbhoir@umass.edu), Shravanthi Murugesan (smurugesan@uma |
| 15 | Text augumentation using LLMs | Sejal Agarwal (sejalagarwal@umass.edu), Siddharth Jain (siddharthjai@umass.edu) |
| 18 | In-context learning for VLM | Abhishek Sureddy (asureddy@umass.edu), Akshay Kumar Sureddy (akshaykumars@u Durga Sandeep Saluru (dsaluru@umass.ec |
| 24 | MindSLM: Fine-tunning SLM's for effective and confidential mental health therapy | Matt Lips (mlips@umass.edu), Xingyu Bian (xingyubian@umass.edu), Zhiyang Zuo (zzuo@umass.edu) |
| 27 | automated code anomaly detection for enhancing software quality | Nikhil Anand (nikhilanand@umass.edu), Rakshita Srivastava (rakshitasriv@umass.e |
| 29 | RL driven portfolio optimization | Aadam Lokhandwala (alokhandwala@umas Kirat Arora (kiratarora@umass.edu), Rohit Goli (rgoli@umass.edu) |
| 35 | Curriculum learning methods benchmarking | Andre Kenneth Chase Randall (andrekenne Joseph Collins (jccollins@umass.edu) |
| 38 | Accelerating TDDFT simulations using LSTM | Dongming Li (dongmingli@umass.edu) |
| 41 | Temporal crowd flow classification of sequntial frames | Vaishnavi Panchavati (vpanchavati@umass Venkata Samyukta Malapaka (vmalapaka@ Yogeshwar Pullagurla (ypullagurla@umass. |
| 48 | Compact diffusion model for cifar10 | Chuchen Li (chuchenli@umass.edu), Isaac Zhong (bzhong@umass.edu) |
| 52 | Classroom learning with knowledge distillation | Matthew Peters (matthewpeter@umass.edu Neeladri Bhuiya (nbhuiya@umass.edu), Shreyaa Dani (sdani@umass.edu) |

| | | |
|---|---|---|
| 53 | autoregressive action generation for diffusion policy | Anushka Agarwal (anushkaagarw@umass. Muhammad Yusuf Hassan (mdhassan@um Talha Mohammed Zakir Chafekar (tchafeka |
| 55 | Knowledge distillation for efficient neural network compression | Akshata Sangwai (asangwai@umass.edu), Prateek Bhindwar (pbhindwar@umass.edu) |
| 60 | compact diffusion models: from symmetry awareness to Knowledge distillation | Ajit Kumar (ajitkumar@umass.edu), Pronay Dutta (pronaydutta@umass.edu) |
| 62 | emotion based style transfer | Gehao Zhang (gehaozhang@umass.edu), Zekai Zhang (zekzhang@umass.edu) |
| 65 | enhancing text classification with llm driven naugumentation for imbalanced datasets | Chandana Pamidi (cpamidi@umass.edu), Mahima Choudha (mchoudha@umass.edu) Ujwala Munigela (umunigela@umass.edu) |
| 66 | can large multimodal models really understand affordance? | Delin Chen (delinchen@umass.edu), Fengming Shen (fengmingshen@umass.ed Siyuan Cen (scen@umass.edu) |
| 70 | evalaution methods for neural style tranfer | Benjamin Hall (bmhall@umass.edu), David Gerard (dgerard@umass.edu) |
| 71 | textual augumentation for medical transcriptions (?) | Shreya Balakrishna (shreyabalakr@umass. Vaishnavi Kashyap (vaishnavikas@umass. |
| 77 | Geo localization [changed topic from "ARC challnege/symbolic learners]" | Hung Nguyen (huntnguyen@umass.edu), John Steenbruggen (jsteenbrugge@umass. Long Vo (longvo@umass.edu) |
| 80 | badminton shot and player movement prediction | Anmol Chokshi (achokshi@umass.edu), Kavisha Parikh (kavishaprana@umass.edu Rahasya Barkur (rbarkur@umass.edu) |
| 83 | Immersive audio | Lixing Fang (lixingfang@umass.edu), Qinhong Zhou (qinhongzhou@umass.edu), Sunli Chen (sunlichen@umass.edu) |
| 85 | automatic speech recognition | Jiageng Liu (jiagengliu@umass.edu), Wenjun Liu (wenjunliu@umass.edu), Zhehuan Chen (zhehuanchen@umass.edu |
| 88 | attribute driven personre-id for passenger counting using sptio-temporal patterns in data scarce scenarios | Nthenya Kyatha (mkyatha@umass.edu), Sanuratu Koroma (skoroma@umass.edu) |

# Zero-shot knowledge graph embedding

Mingchen Li (mingchenli@umass.edu)
Rohan Pandey (rohanpandey@umass.edu)

Project #1

# Motivation and background

## Motivation:

Knowledge graphs (KG) such as Wikidata and Free- base are an important resource for many applications of artificial intelligence.  However, it is impractical to include all types of relations in knowledge graph applications. Zero-shot learning (ZSL) for knowledge graph embedding was thus introduced to deal with unseen relations which are not available during training.

***Neighbor relations are a crucial external source for zero-shot knowledge graph embedding*** *because they not only provide prior knowledge of a relation when transferred to unseen relations but also help the knowledge graph embedding model reduce the size of candidate entities, thus improving the performance of related tasks, such as link prediction.*

## Previous Work:

***Previous works utilize the internal text description of relations*** and external structure information provided by the ontology or the knowledge graph, but they ignore the extra structural knowledge of the neighbor relations that may further promote zero-shot learning performance. ***We argue that neighbor relations are a crucial external source*** for zero-shot knowledge graph embedding because they not only provide prior knowledge of a relation when transferred to unseen relations but also help the knowledge graph embedding model reduce the size of candidate entities, thus improving the performance of related tasks, such as link prediction and question answering.

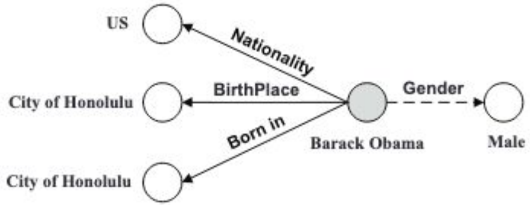



Figure 1.  A sub-graph of the entity *Barack Obama*.  The objective relation `Gender` has the neighbor relations `Nationality`, `BirthPlace` and `Born in`.

Following previous work on link prediction [8, 15], we will evaluate the ZSL Link Prediction task using the Mean Reciprocal Rank (MRR), hits@*10*, hits@*5* and hits@*1*, and two datasets NELL-ZS, Wiki-ZS.

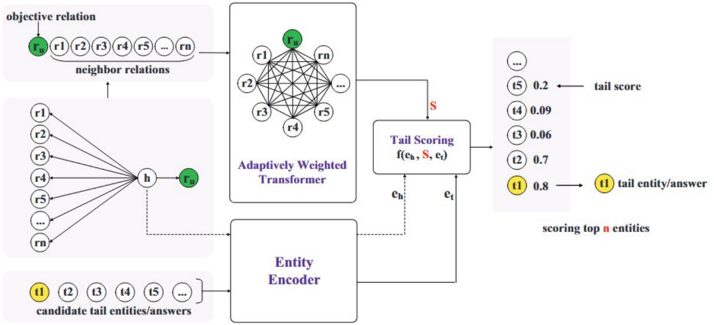
| Dataset | #Entities | #Triples | #Train | #Dev | #Test |
|---|---|---|---|---|---|
| NELL-ZS | 65,567 | 188,392 | 139 | 10 | 32 |
| Wiki-ZS | 605,812 | 724,967 | 469 | 20 | 48 |

Table 1. Statistics of the NELL-ZS and Wiki-ZS

# Results

## Method A: Training Adaptive Transformer

| KGE model | Method | NELL-ZS MRR | NELL-ZS hits@10 | NELL-ZS hits@5 | NELL-ZS hits@1 | Wiki-ZS MRR | Wiki-ZS hits@10 | Wiki-ZS hits@5 | Wiki-ZS hits@1 |
|---|---|---|---|---|---|---|---|---|---|
| TransE | ZSL-TransE | 0.097 | 0.203 | 0.147 | 0.043 | 0.053 | 0.119 | 0.081 | 0.018 |
| | Cosine-Map | 0.097 | 0.212 | 0.134 | 0.042 | 0.084 | 0.198 | 0.109 | 0.036 |
| | OntoZSL[10] | 0.250 | 0.399 | 0.327 | 0.172 | 0.184 | 0.265 | 0.215 | 0.138 |
| | ZSGAN[21] | 0.234 | 0.373 | 0.304 | 0.160 | 0.177 | 0.258 | 0.207 | 0.131 |
| | LENR | **0.271** | **0.403** | **0.344** | **0.200** | **0.218** | **0.286** | **0.254** | **0.179** |
| DistMult | ZSL-DistMult | 0.235 | 0.326 | 0.284 | 0.185 | 0.189 | 0.236 | 0.210 | 0.161 |
| | Cosine-Map | 0.088 | 0.179 | 0.111 | 0.045 | 0.089 | 0.197 | 0.107 | 0.040 |
| | OntoZSL[10] | 0.256 | **0.385** | 0.318 | 0.188 | 0.211 | **0.289** | 0.238 | 0.167 |
| | ZSGAN[21] | 0.249 | 0.376 | 0.306 | 0.183 | 0.207 | 0.284 | 0.235 | 0.164 |
| | LENR | **0.270** | 0.382 | **0.330** | **0.206** | **0.212** | 0.275 | **0.238** | **0.174** |
| Tucker | ZSL-Tucker | 0.253 | 0.382 | 0.317 | 0.185 | 0.109 | 0.225 | 0.189 | 0.048 |
| | Cosine-Map | 0.169 | 0.296 | 0.236 | 0.100 | 0.111 | 0.227 | 0.158 | 0.052 |
| | OntoZSL[10] | 0.208 | 0.298 | 0.248 | 0.158 | 0.076 | 0.180 | 0.100 | 0.024 |
| | ZSGAN[21] | 0.193 | 0.309 | 0.247 | 0.132 | 0.160 | 0.240 | 0.206 | 0.115 |
| | LENR | **0.283** | **0.406** | **0.356** | **0.214** | **0.199** | **0.255** | **0.228** | **0.164** |




## Method B: Inference only LLM

| Model | Method | NELL-ZS MRR | NELL-ZS hits@10 | NELL-ZS hits@5 | NELL-ZS hits@1 |
|---|---|---|---|---|---|
| Llama-3.1-7B | Zero-Shot | 0.1185 | 0.1021 | 0.1406 | 0.1765 |
| | Neighbour Enhanced (Using relations from Training Method) | 0.1178 | 0.1028 | 0.14 | 0.1734 |
| | Relation Based Iteration (Our Method) | **0.3419** | **0.3299** | **0.3549** | **0.3799** |

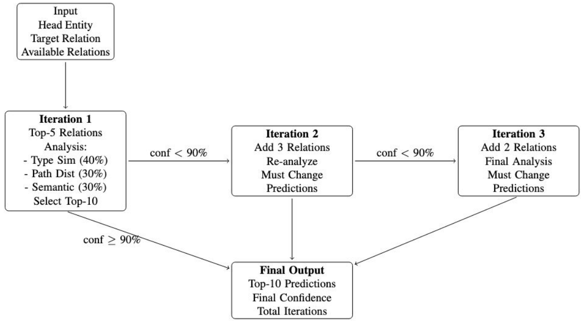



Figure 4. Iterative Relation Prediction Method

# Conclusion

In our work, we have ***proposed a novel neighbour enhanced training methodology for zero-shot link prediction tasks***. In addition, we demonstrated that modern ***LLMs with their pre-training capacity perform at least as well as training based methods using just inference time methods.***

Our results demonstrate that ***neighbour-enhanced methods perform better across the board***. For training based methods, our proposed neighbour enhanced methodology performs better than existing methods. Furthermore, our novel prompting/inference based strategy drastically improves performance baseline zero-shot and neighbour enhanced methods, thereby showing the potential of LLMs and test-time scaling methods.

Due to a lack of computation, ***our work is currently limited to only a single dataset as well as a single model.*** In future work, we would like to extend the analysis across more datasets, and models to demonstrate the generalizability of our results. In addition, ***for inference based scaling, we have currently restricted the maximum iterations for 3,*** we would like to extend this to more iterations to see if/when does the performance gains saturate. We would also like to ***compare the performance with fine-tuning*** of the language models.

# Medical image anomaly detection using VLM

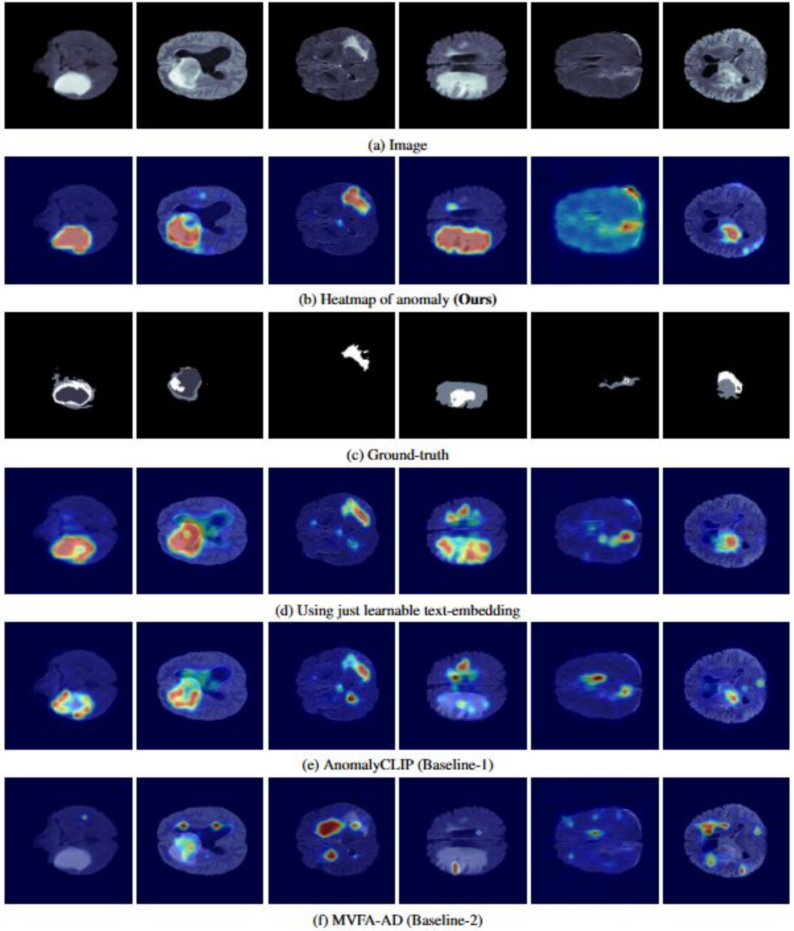
Deepti Guntur (dguntur@umass.edu),
Lucy Bodtman (lbodtman@umass.edu)

Project #9

# MedVMAD

***Medical imaging Anomaly Detection using Visual Language Models***

- **Motivation:**
  - Subtle abnormalities in medical imaging are often missed, impacting diagnostic accuracy.
- **Goal:**
  - Develop a framework using the visual language model, CLIP, for anomaly detection in medical images.
- **Focus:**
  - Anomaly detection using learnable embeddings for both text prompts and images.
  - Trained on brain MRI scans, tested on brain MRI scans, liver CT-scans, and breast cancer tomosynthesis for zero-shot testing.
- **Background:**
  - AnomalyCLIP - uses only learnable text embeddings
  - MVFA-AD - uses only learnable visual embeddings
  - MediCLIP - Few-shot AD with learnable embeddings



(a) Image

(b) Heatmap of anomaly **(Ours)**

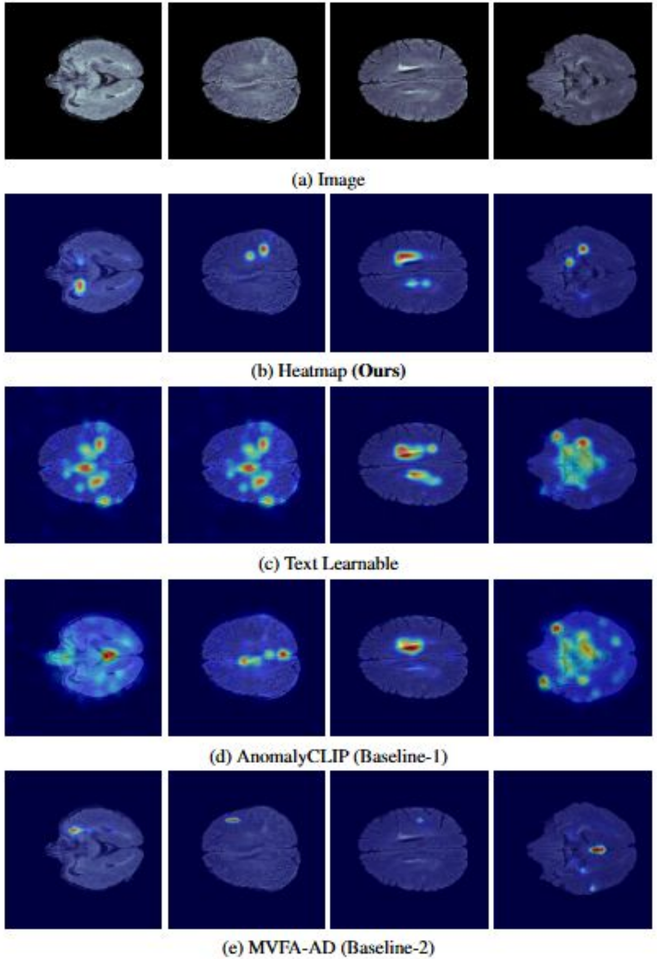
(c) Ground-truth

(d) Using just learnable text-embedding

(e) AnomalyCLIP (Baseline-1)

(f) MVFA-AD (Baseline-2)

Figure 1. Results for anomalous images



(a) Image

(b) Heatmap **(Ours)**

(c) Text Learnable

(d) AnomalyCLIP (Baseline-1)

(e) MVFA-AD (Baseline-2)

Figure 2. Results for non-anomalous images

# Method & Conclusion

- We utilize image feature token embeddings and textual embeddings to compute similarity scores, while segmentation token embeddings and ground-truth annotations are used to generate anomaly maps. These two sets of scores are combined to define our loss function..
- Evaluation Metrics with BRaTS 2021 dataset:

| Model | Pixel AUROC | Pixel AUPRO | Image AUROC | Image AP |
|---|---|---|---|---|
| AnomalyCLIP | 96.5 | **77** | 70.3 | 76 |
| MVFA-AD | 89.1 | 57.4 | 79.6 | 84.5 |
| MedVMAD (Ours) | **96.7** | 56.9 | **84.6** | **87.7** |

- Incorporating both learnable image embeddings and learnable text embeddings significantly enhances the precise anomaly detection compared to using just learnable text embeddings alone.

# Domain adaptation for OCR models

Frankie Furnelli (ffurnelli@umass.edu),
**Thomas Potts (tpotts@umass.edu)**,
Virtulya Rajput (vrajput@umass.edu)

Project #12

# Overview, Motivation and Background

**Project Overview:**

- To adapt the TrOCR model to better recognize an unseen handwritten domain:
  - A dataset of personalized journal entries that we collected, cleaned and preprocessed ourselves.
  - 197 pages / 6,595 lines of handwritten text
- Domain Adaptation Techniques:
  - Pseudo-labeling and Masking
  - Seeding a limited amount of manually labeled data

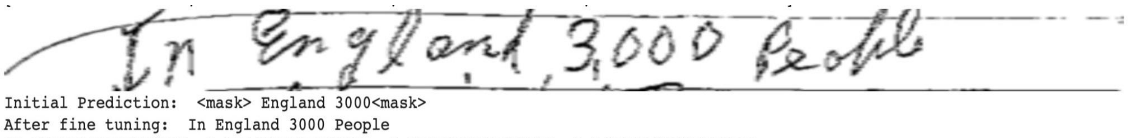
**Motivation:**

- How to improve pre-trained OCR models when labeled documents are scarce

**Background:**

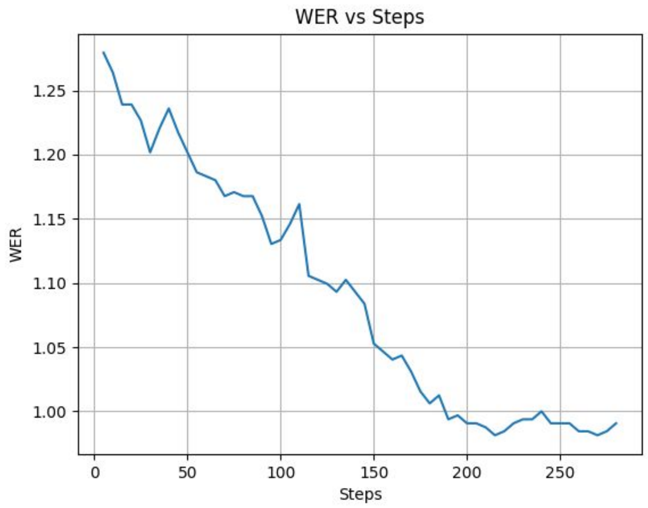
- TrOCR is a state-of-the-art Transformer-based OCR model that is effective for standard handwriting and printed text.
- TrOCR struggles to correctly recognize unseen personal writing of a single author with unique style and quirks.
- We seek to reduce this gap by fine-tuning the model with unlabeled handwriting samples for domain adaptation.

# Results




Example of initial TrOCR pseudo-label vs. fine-tuned prediction

| Model | CER | WER |
|---|---|---|
| TrOCR-SMALL | 0.7649 | 1.2727 |
| TrOCR-SMALL-handwritten | 0.7920 | 1.3374 |
| Adapted TrOCR with pseudo-labels | 0.6049 | 1.0461 |
| Fully Adapted TrOCR | 0.5890 | 0.9907 |

# Conclusion

The adaptation methods, pseudo-labeling and training on limited annotated data, improved the CER and WER of TrOCR on the personalized handwritten domain!

**Applications:**

- There is a plethora of documents that have not been transcribed, such as historical documents and personal notes or journals
- Implementing methods to adapt OCR models to unlabeled or partially labeled data may prove useful for historical research and digitizing notes and journals.

**Limitations:**

```
['_a', '_b', '_c', '_d', '_e', '_f', '_g', '_h', '_i', '_j', '_k', '_l', '_h', '_i']
['a b c d e f g h i j k l h i']
```

- Model Hallucination:
  - TrOCR was likely to produce long false predictions when preprocessing was imperfect, such as text data being cut off or noise in an image
- Preprocessing:
  - TrOCR requires text images to be only one line. The handwritten dataset lacked of structure, which caused it to be challenging to separate the data into neat lines of text
    - We utilized OpenCV to detect horizontal lines, and partitioned from the predictions

# Optimizing Few-Shot Learning: A Semi-supervised hybrid approach for enhanced classification on Omniglot

Rutika Bhoir (rbhoir@umass.edu),
Shravanthi Murugesan (smurugesan@umass.edu)

Project #13

# Optimizing Few-Shot Learning: Semi-Supervised Hybrid Approach for Enhanced Classification on Omniglot

## Summary of the Project

- **Hybrid Approach:** Combines semi-supervised learning with Model-Agnostic Meta-Learning (MAML).
- **Objective:** Address data scarcity by generating pseudo-labels for unlabeled data, effectively expanding the training set.
- **Dataset:** Utilizes the Omniglot dataset, with over 1,600 characters across 50 alphabets, challenging models to generalize with minimal labeled examples.

- **Motivation:**

Few-shot learning enables models to generalize effectively to unseen classes with minimal labeled data.

Crucial for domains with limited labeled data availability, such as medical diagnostics and specialized research fields

- **Proposed Approach:**
  - Use pseudo-labels to expand the training set with high-confidence samples.
  - Integrate augmented data into MAML to enhance model adaptability and accuracy

**Goal:**

- Develop robust, generalizable models capable of addressing real-world challenges in few-shot learning.

## **Previous Work**:

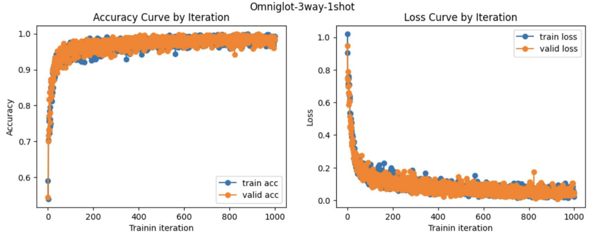
Earlier methods in few-shot learning rely heavily on using embeddings or structured knowledge like class hierarchies to generalize from limited data. While effective, they miss out on using additional information, like unlabeled data, that could boost performance.

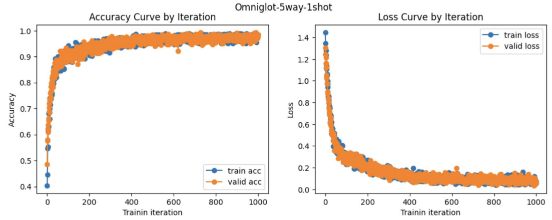
Approaches like Matching Networks and Prototypical Networks work well but struggle with tasks involving high variation or imbalanced data.

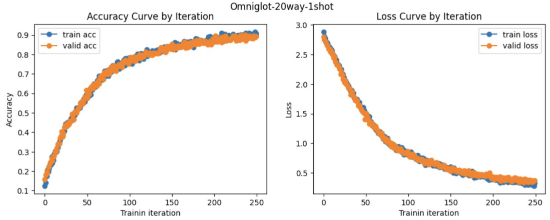
Similarly, MAML, while flexible, doesn't utilize unlabeled data, leaving room for improvement.By integrating semi-supervised learning with pseudo-labeling, we can expand the training set and improve the model's ability to generalize, addressing these gaps in existing methods.

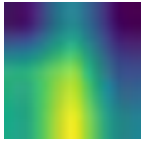
## **Results**

- **MAML Performance**:
  **5-way 1-shot**: 95.32% accuracy after 1000 iterations.
- **3-way 1-shot**: 97.08% accuracy after 1000 iterations.
- **20-way 1-shot**: 82.99% accuracy after 250 iterations.
- **Hybrid Approach with Pseudo-Labels**:
  **5-way 1-shot**: 95.39% accuracy.
- **3-way 1-shot**: 97.10% accuracy.
- **20-way 1-shot**: 83.47% accuracy.

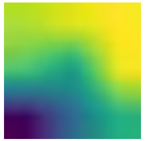


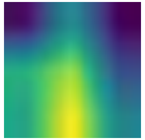



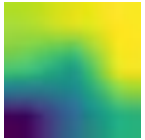
MAML



Hybrid



MAML



Hybrid

# CONCLUSION

Our experiments demonstrated the effectiveness of the hybrid semi-supervised learning approach combined with Model-Agnostic Meta-Learning (MAML) for few-shot learning tasks on the Omniglot dataset. The results indicate that incorporating pseudo-labeled data enhances generalization and model adaptability as seen in 20-way one-shot task.

1. **5-Way 1-Shot Classification**: Achieved a test accuracy of 95.32% over 1000 iterations, showcasing strong performance in a limited-label setting.
2. **3-Way 1-Shot Classification**: Delivered competitive accuracy with consistent training and validation trends, reflecting the model's robustness.
3. **20-Way 1-Shot Classification**: Despite the increased complexity, the model attained an accuracy of 83.47% in just 250 iterations, demonstrating its scalability and efficiency.

**The training and validation curves highlight minimal overfitting across all tasks, with rapid convergence within the initial iterations. These findings underscore the potential of our approach to address data scarcity and task adaptability challenges in few-shot learning scenarios. Future work will aim to further optimize computational efficiency and explore broader applications in real-world datasets beyond Omniglot.**

# Text augmentation using LLMs

**Sejal Agarwal** (sejalagarwal@umass.edu),

Siddharth Jain (siddharthjai@umass.edu)

Project #15

# Problem Statement:

The goal of this project is to explore the effectiveness of Large Language Models (LLMs) like GPT-2 for augmenting imbalanced datasets in text classification. We aim to improve model performance, particularly for minority classes, by generating synthetic data through context-aware prompts in various imbalance scenarios.

# Background:

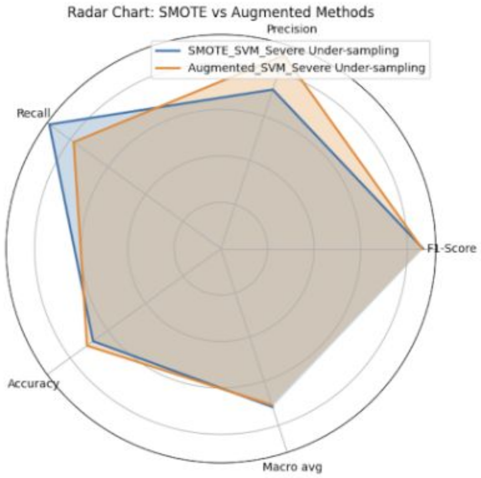
Imbalanced datasets are a common challenge in text classification, often resulting in biased models that perform poorly on underrepresented classes. This issue is particularly critical in real-world applications like healthcare, finance, and social media, where minority classes hold significant value. Traditional methods like SMOTE may not fully address the complexities of imbalances in text data.

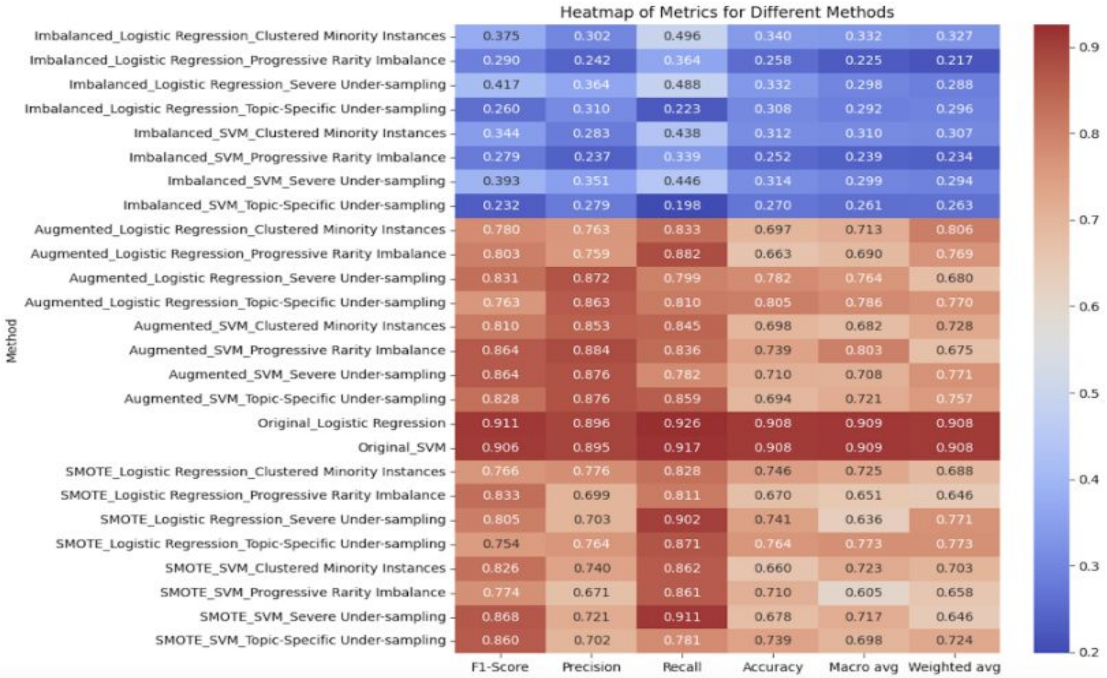
# Method:

- **Data:** Used AG News dataset with imbalance in the Science/Technology category.
- **Imbalance Technique:** Applied severe under-sampling, topic-specific sampling, clustered minority sampling, and progressive rarity imbalance—to simulate real-world imbalances.
- **Baseline Model:** Used the Synthetic Minority Oversampling Technique (**SMOTE**) for comparison, generating synthetic instances for the minority class.
- **LLM:** Used DistilGPT-2-based Large Language Models (**LLMs**) with context-aware prompts to generate synthetic text for underrepresented classes, enhancing the dataset.
- **Model Training:** Train Logistic Regression and Support Vector Machine models, evaluating performance on metrics like accuracy, precision, recall, and F1-score, with a focus on minority class recall.

# Result:

- Augmented methods outperform SMOTE across all metrics in severe under-sampling scenarios (radar chart).
- Original models perform best on balanced datasets but struggle with imbalance (heatmap).
- Augmented approaches improve F1-score and recall significantly, surpassing SMOTE.
- Imbalanced methods without augmentation have the weakest performance.




# Conclusion:

The findings suggest that LLM-based augmentation offers notable strengths, such as superior recall and the generation of diverse, contextually relevant synthetic samples. However, traditional methods like SMOTE, while effective in simpler cases, struggle to capture more complex patterns, limiting their ability to address class imbalances.

# In-context learning for VLM

**Abhishek Sureddy (asureddy@umass.edu),**
Akshay Kumar Sureddy (akshaykumars@umass.edu),
Durga Sandeep Saluru (dsaluru@umass.edu)

Project #18

# Motivation and background

## Motivation

- **ICL Success in NLP**: Adapts to tasks without fine-tuning; reduces computational overhead.
- Vision tasks pose unique challenges (data structures, alignment with ICL principles).
- Potential to enhance **flexibility** and **generalization** of Vision-Language Models (VLMs).

## Background

- Vision-Language Models (e.g., **Flamingo**, **ViLA**) process interleaved image-text sequences, enabling multi-example prompts.
- Challenges: Most VLMs trained on single image-text inputs; adapting them for ICL is non-trivial.
- Leverage **ViLA models** as they support multi-example inputs, making them feasible for ICL.

- Investigate how **In-Context Learning (ICL)**, proven effective in language models (e.g., GPT), can be applied to **vision tasks**.
- Evaluate performance on vision tasks: **Image Captioning**, **Visual Question Answering (VQA)**, **Classification** (coarse and fine-grained), and **Keypoint Detection**.
- Analyze the impact of **quantity** and **quality** of in-context examples on performance using **ViLA models (3B, 13B), using 2 settings**:
  1. Random retrieval
  2. Nearest Neighbor Search Retrieval

# Methodology & Results

## Coarse-classification Prompt: Imagenet

<image> This image shows a [class-1]<|endofchunk|>
<image> This image shows a [class-2]<|endofchunk|>
<image> This image shows a

Random setting - Model output: cat

Query Image

potpie | pick | siamese cat

NNS setting - Model output: siamese cat

Query Image

pekinese | persian cat | siamese cat

## Finegrain-classification Prompt: Stanford cars

Identify and classify the car in the provided image. Provide the label in the exact format: [Make] [Model] [Year].
<image> [class-1]<|endofchunk|>
<image> [class-2]<|endofchunk|>
<image>

Random: Chevrolet Impala Corvette Stingray Coupe 2012

Query Image

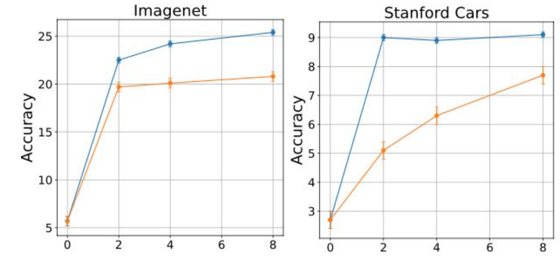
Dodge Durango SUV 2007 | Toyota Camry Sedan 2012 | Chevrolet Impala Sedan 2007

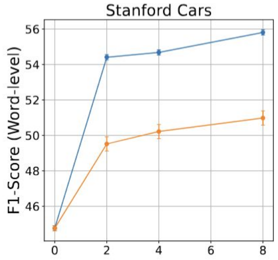
NNS: Chevrolet Impala Sedan 2007

Honda Odyssey Minivan 2007 | BMW 6 Series Convertible 2007 | Chevrolet Impala Sedan 2007

# Results & Conclusion

## Classification task

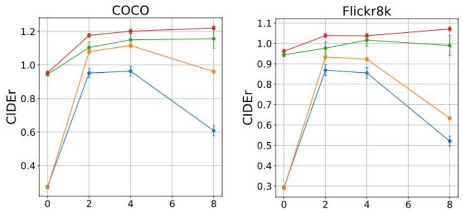


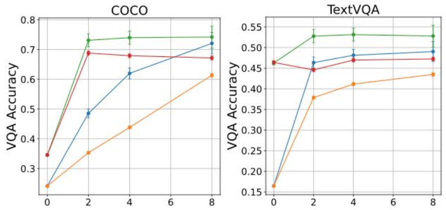


Results for OF-3b model
(metric Vs n-shot)

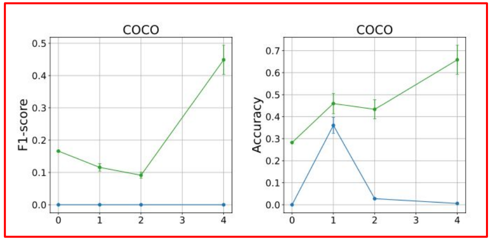
## Other tasks



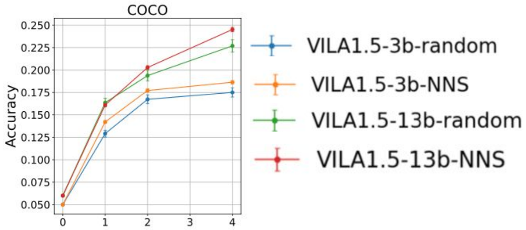

Captioning Task




VQA Task




Keypoint Tasks




- Generally, as n-shots increases, performance increase (small/ large model)
- Except for VQA, NNS performs better than random setting in all other tasks
  => VQA needs diversity (need more experiments to confirm)
- Larger model could have Emergent properties in its incontext ability.
  (ViLA-13b, sudden performance jump from 2 shot to 4 shot in keypoint task.

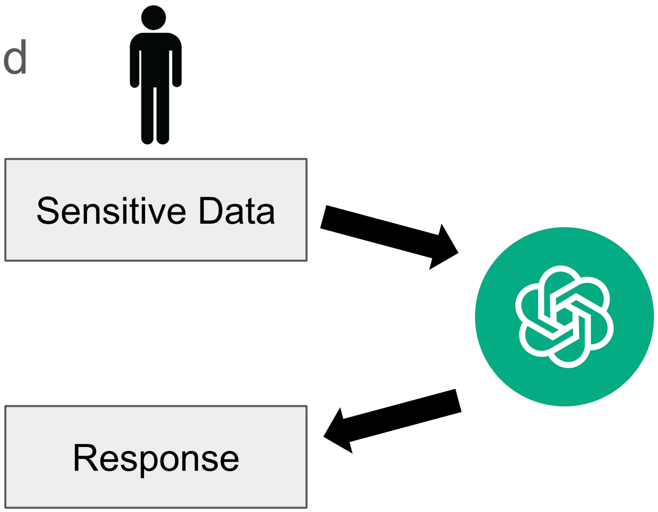
# MindSLM: Fine-tuning SLMs for effective and confidential mental health therapy

Matt Lips (mlips@umass.edu),
Xingyu Bian (xingyubian@umass.edu),
Zhiyang Zuo (zzuo@umass.edu)

Project #24

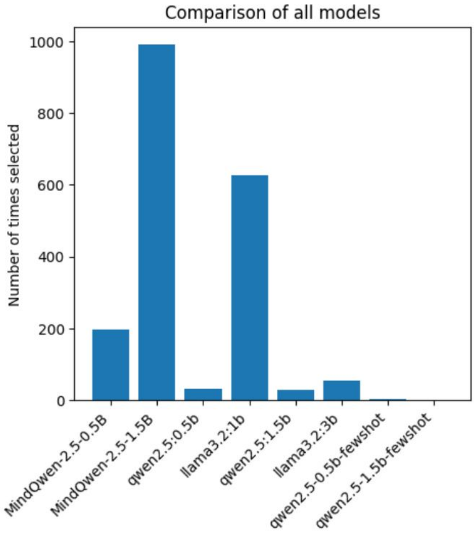
# Quick summary of the project, motivation and background

- LLMs have great potential in the medical field
- LLMs
  - Powerful
  - Closed source (OpenAI)
  - Requires a lot of compute
  - Needs remote access
- Existing smaller language models
  - Less optimal performance
  - Open source
  - Run locally on weaker hardware
- Mental health epidemic
- Widely accessible mental health support
- Fine-tune smaller models on counseling dataset and few-shot prompting



# Main results + comparison to prior work

- Our MindQwen outperformed other models quantitatively through ROUGE-L and BERTScore, and qualitatively through LLM-as-a-judge
- Larger models performed better, few shot improved smaller models but not larger models
- First to utilize our quantitative metrics in mental health

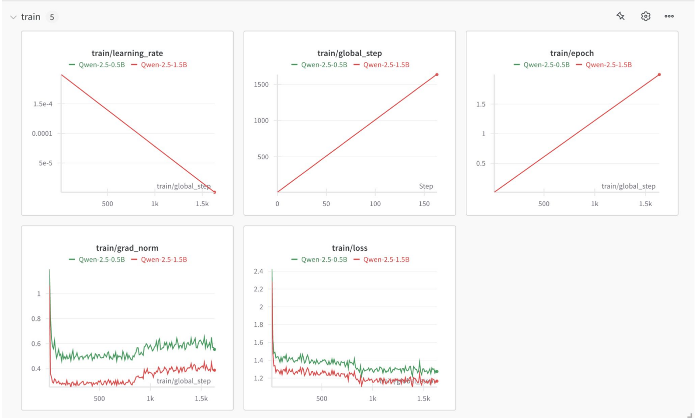



| Model Name | # outputs | percentage |
|---|---|---|
| qwen2.5:0.5b-instruct | 24 | 1.5% |
| llama3.2:1b-instruct | 1143 | 70.0% |
| qwen2.5:1.5b-instruct | 216 | 13.2% |
| llama3.2:03b-instruct | 251 | 15.4% |

| Model Name | # outputs | percentage |
|---|---|---|
| qwen2.5:0.5b-instruct | 319 | 19.7% |
| MindQwen2.5:0.5b | 1299 | 80.2% |
| qwen2.5:0.5b-fewshot | 2 | 0.1% |

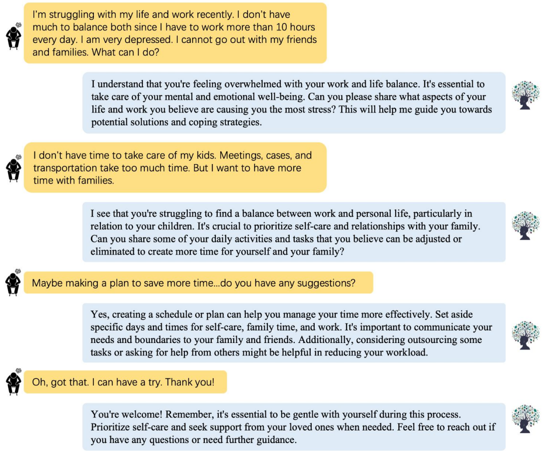
| Model Name | # outputs | percentage |
|---|---|---|
| qwen2.5:1.5b-instruct | 524 | 32.0% |
| MindQwen2.5:1.5b | 1114 | 68.0% |
| qwen2.5:1.5b-fewshot | 0 | 0.0% |

| Model Name | # outputs | percentage |
|---|---|---|
| MindQwen2.5:0.5b | 606 | 38.3% |
| MindQwen2.5:1.5b | 976 | 61.7% |



# Conclusion

- Our model performed best based on our results and analysis
- Indicates potential use case for our product in the real world and improve mental health well being
- Limitations
  - Llama-3.2 training faced bug in tokenization
- Future work
  - Newer models
  - Multi-turn conversations
  - Agentic workflows

# Automated code anomaly detection for enhancing software quality

Nikhil Anand (nikhilanand@umass.edu),
**Rakshita Srivastava (rakshitasriv@umass.edu)**

Project #27

# Summary and Motivation

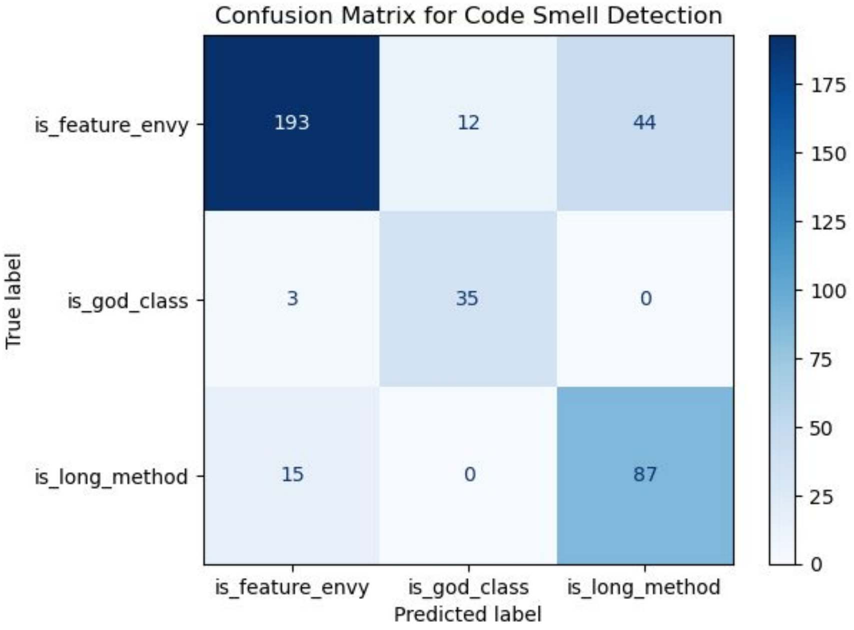
### Summary

This project leverages LSTM networks and CK metrics to detect code anomalies, such as defects and design flaws, by analyzing labeled code samples. It aims to enhance software quality and explores automated refactoring suggestions to improve maintainability.

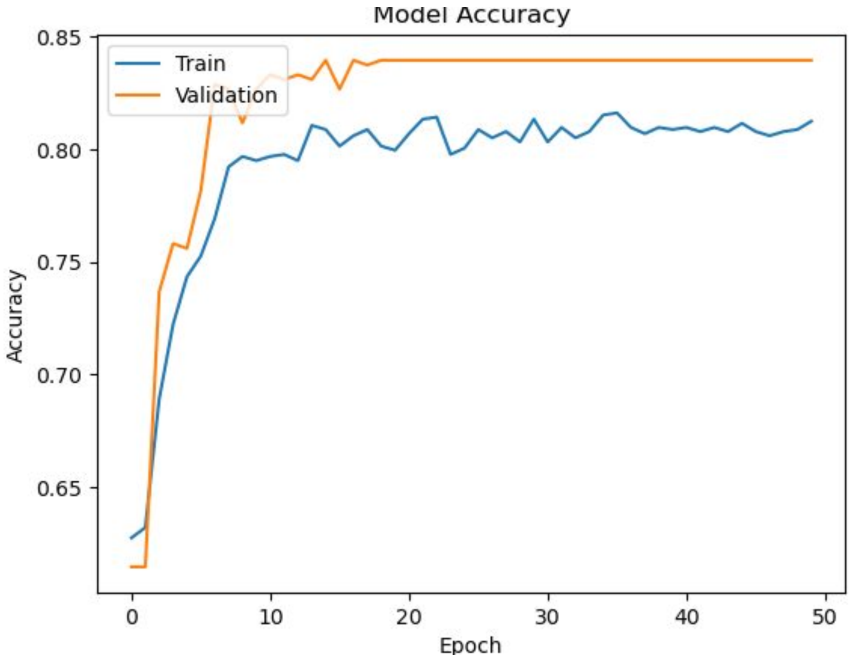
### Motivation

1. Improving Software Quality and Maintainability
2. Empowering Developers with Actionable Insights
3. Advancing the Role of AI in Software Engineering

# Results




Confusion Matrix




Training vs Validation

# Conclusion

1. Effectiveness of LSTM Models
2. Superiority Over Dense Neural Networks (DNNs)
3. Broader Implications for Software Quality
4. Challenges in Precision for Certain Code Smells
5. Future Directions

Project #29

# Reinforcement Learning-Driven Portfolio Optimization

## An LSTM-Enhanced PPO Network Approach for Subsector trading in the S&P 500

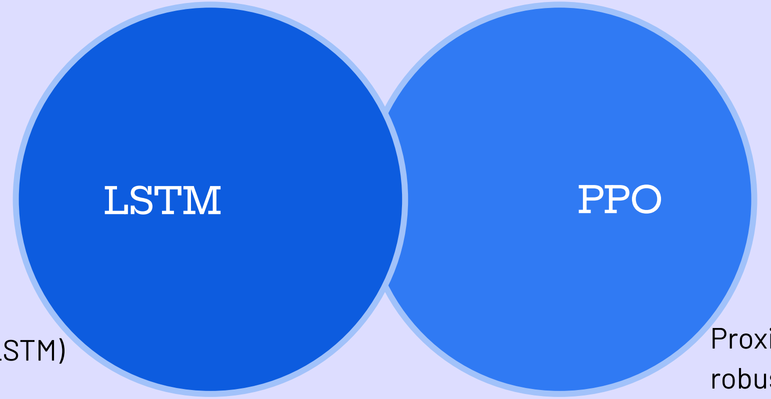
**Aadam Lokhandwala**
alokhandwala@umass.edu

Rohit Goli
rgoli@umass.edu

Kirat Arora
kiratarora@umass.edu

# Challenges in Portfolio Management

- Traditional strategies are static and fail to adapt to market volatility.
- So our **goal** is to develop a dynamic model that learns optimal allocation across sub sectors using temporal market data.




The Long Short-Term Memory (LSTM) component captures temporal dependencies in financial time series data, addressing challenges posed by non-stationary and autocorrelated market data.

Proximal Policy Optimization (PPO) is a robust and efficient approach to optimizing trading strategies by directly learning the policy that maximizes expected returns while managing risk. It is useful because of its ability to handle continuous action spaces and maintain stability during training

# Data Collection And Environment

**Data Collection:** We have assembled a comprehensive dataset with 570+ Technical indicators (e.g., SMA, RSI, MAMA, etc.) spanning from January 1, 2000, to 2019 for training, with data from 2020 onwards reserved for testing. This data was collected using Alpha Vantage API, with some of it being self computed.

**Data Preprocessing:**

- Inter-Day percentage change between features.
- Missing data imputation.

**State Space:**

- **Features:** Hidden feature from LSTM, PCT OHLCV, and PCT technical indicators .
- **Portfolio State:** Current portfolio, hidden feature from LSTM.

**Action Space:**

- Continuous allocation weights across subsectors.

**Reward Function:**

$$D_t = \frac{\partial U_t}{\partial \eta} = \frac{B_{t-1}\Delta A_t - A_{t-1}\Delta B_t / 2}{(B_{t-1} - A^2_{t-1})^{3/2}}$$

**; where**

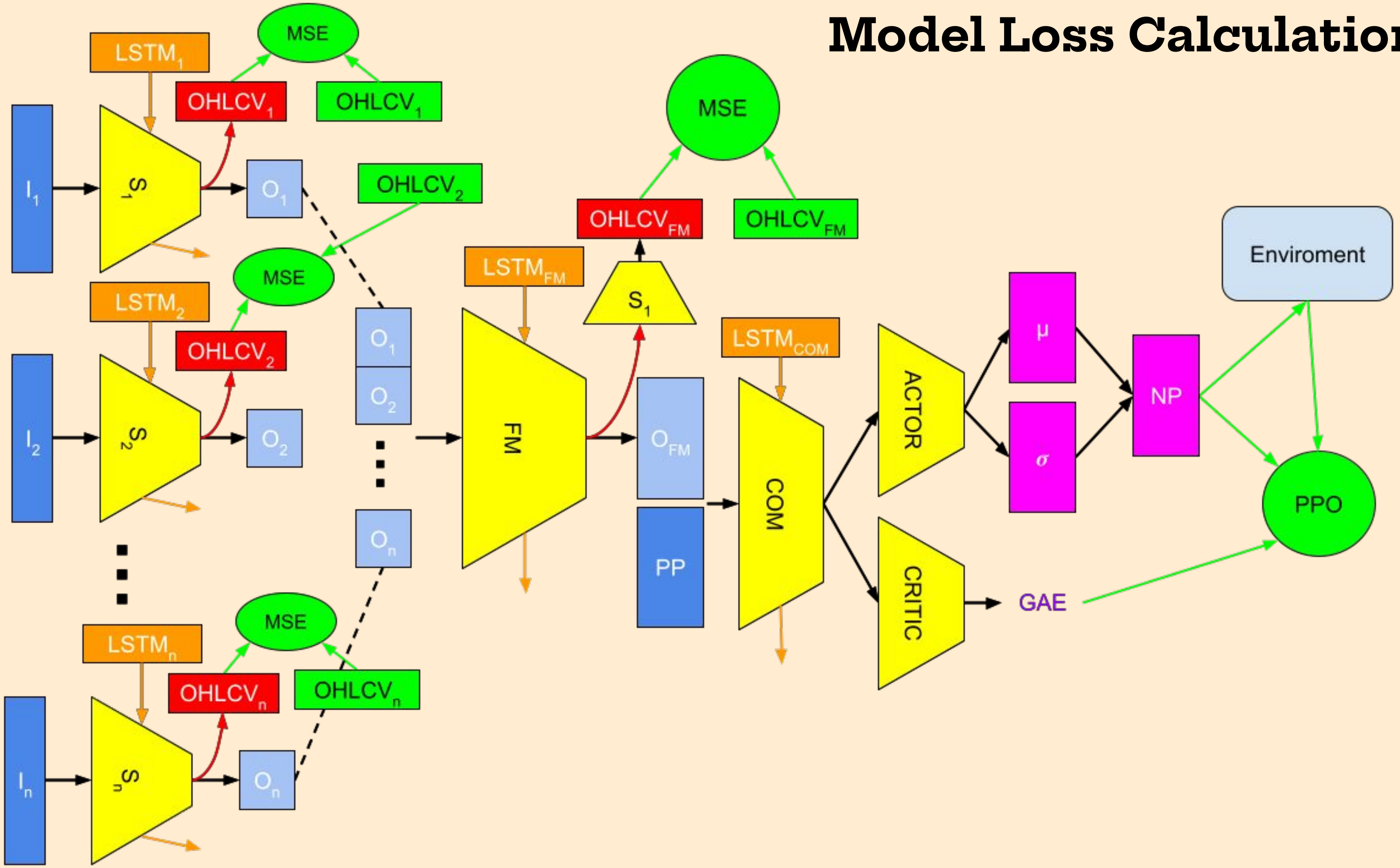
$$A_t = \eta R_t + (1-\eta) A_{t-1} = A_{t-1} + \eta \Delta A_t$$

$$B_t = \eta R_t^2 + (1-\eta) B_{t-1} = B_{t-1} + \eta \Delta B_t$$

**Trading Constraints:**

- Our traded do not affect the market prices.
- Portfolio rebalanced at every market open.
- No trades during the day to simulate day trading and manage computational load.

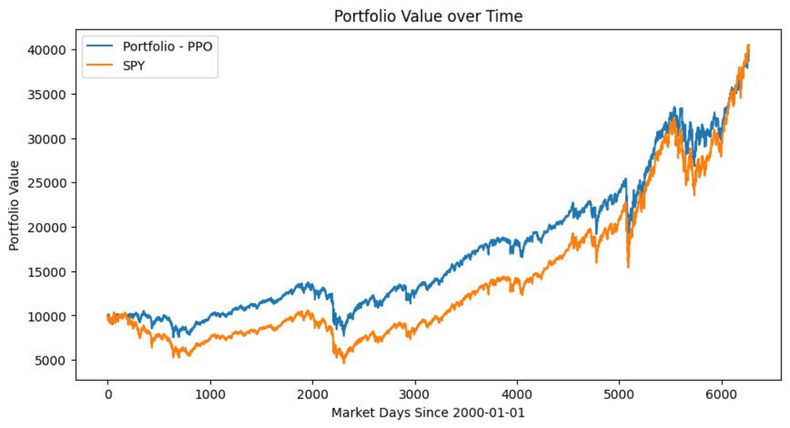
# Model Loss Calculations

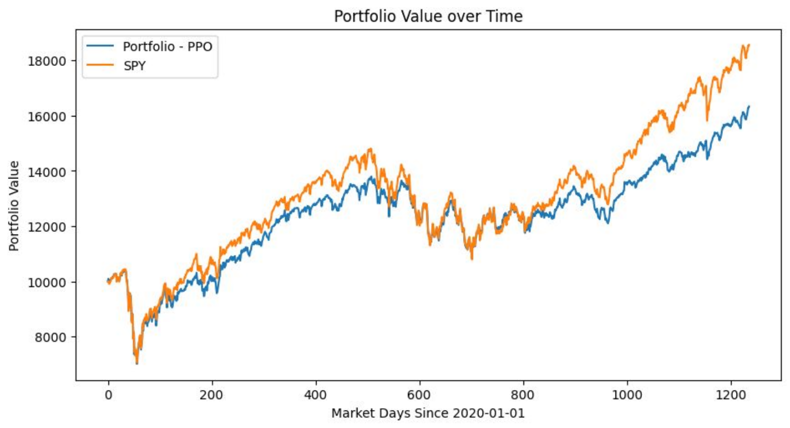


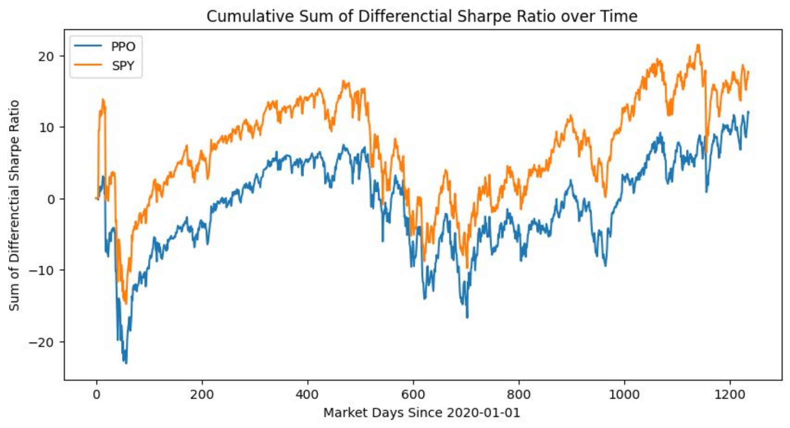
# Results and Conclusion

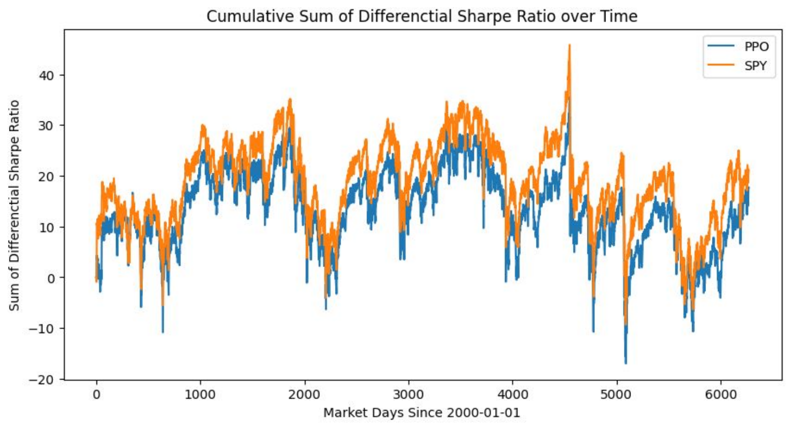
| Article | Data | State Space | Action Space | Reward | Performance |
| --- | --- | --- | --- | --- | --- |
| **Deep Reinforcement Learning Agent for S&P 500 Stock Selection [67]** | 28 S&P500, 5283 days, 415 stocks | Total return, closing price, volumes, quarterly earnings, dividends, declaration and publication dates (d, n, f) | Continuous 20 dims | Differential Sharpe | 1255 days, Total return: 328%, Sharpe = 0.91 |
| **Diversity-driven knowledge distillation for financial trading [68]** | 28 FX, 1h | Normalized OHLCV, current open position as one-hot | Discrete (exit, buy, sell) | PnL | 2 years, PnL $\approx 0.35$ |
| **A Deep RL Approach for Automated Cryptocurrency Trading [69]** | Bitcoin, 3200 h | Time stamp, OHLCV, USD volume weighted Bitcoin price | Discrete (buy, hold, sell) | Profit Sharpe | 800h, return $\approx 3\%$ |
| **An application of DRL to algorithmic trading [39]** | 30 stocks daily, 5 years | Current trading position, OHLCV | Discrete (buy, sell) | Normalized price change | Sharpe = 0.4, AAPL annual return = 0.32 |
| **Adaptive Stock Trading with DRL Methods [63]** | 15 stocks daily, 8 years | OHLCV + Technical indicators (MA, EMA, MACD, BIAS, VR, and OBV) | Discrete (long, neutral, short) | Sortino ratio | 3 years, $-6\%$ to 200% |
| **Portfolio management system in equity market neutral using RL [70]** | 50 stocks daily, 2 years | Normalized OHLC | Continuous 50 dims | Sharpe ratio | 2 years, profit $\approx 50\%$ |
| **DRL for Automated Stock Trading: An Ensemble Strategy [47]** | 30 stocks daily, 7 years | Balance, shares, price, technical indicators (MACD, RSI, CCI, ADX) | Discrete $(2k+1)^{30}$ | Portfolio value change, Turbulence | Sharpe = 1.3, Annual return = 13% |
| **Portfolio trading system for digital currencies with attention gating [71]** | 20 assets, 30 min, 4 years | Normalized HLC, shares | Continuous 20 dims | Log cumulative return | 2 months, Return = 22x |
| **Deep Reinforcement Learning Agent for S&P 500 Stock Selection** | 415 stocks from S&P500 (1998–2018) | Currently available market information (Number of shares, time period, features of stock) | Continuous 20 dims | Differential Sharpe | Sharpe: A = 1.99, B = 2.18, C = 2.20. Total Return: A = 86.38%, B = 82.42%, C = 87.23%. |
| **Our Approach** | Market Indicators and OHLCV for all sectors of S&P 500 | Embedding from the model predicting market data and previous distribution of sectors | Continuous number of sectors + cash | Differential Sharpe | Sharpe Total Return: 12, Portfolio change value 300% (2000 onwards) |

# Results and Conclusion



# Curriculum learning methods benchmarking

Andre Kenneth Chase Randall (andrekenneth@umass.edu),
Joseph Collins (jccollins@umass.edu)

Project #35

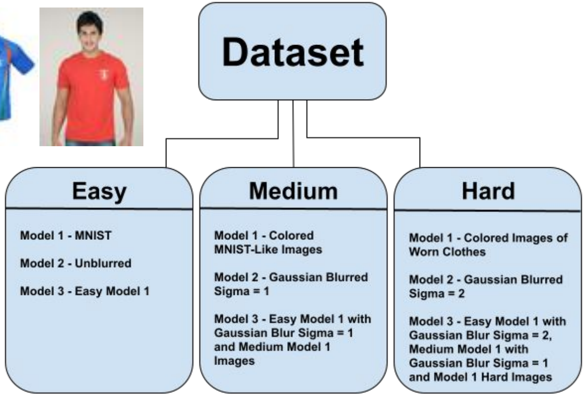
# Introduction and Motivation

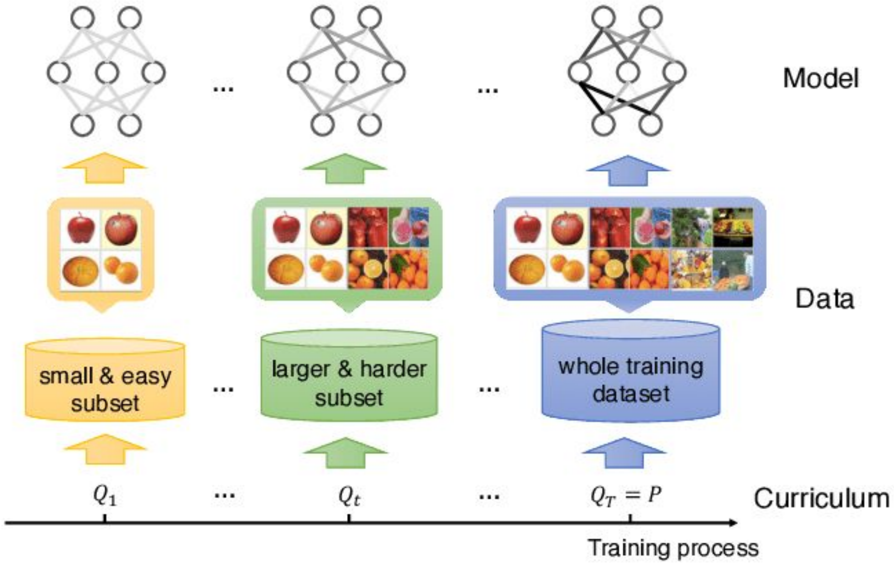
Curriculum learning involves splitting datasets and incrementally training models on the splits.

Aim to explore and benchmark different methods of curriculum learning

Benchmark 3 Main Points

1. Training Time
2. Accuracy on Clean Testing Set
3. Accuracy on Noisy Testing Set




https://www.researchgate.net/figure/llustration-of-the-Curriculum-Learning-CL-concept-The-fruit-images-are-from-106-CL_fig1_350459224

3 Main Methods of Splitting Dataset

1. Complexity of Dataset
2. Amount of Noise in Dataset
3. Hybrid of Previous Methods

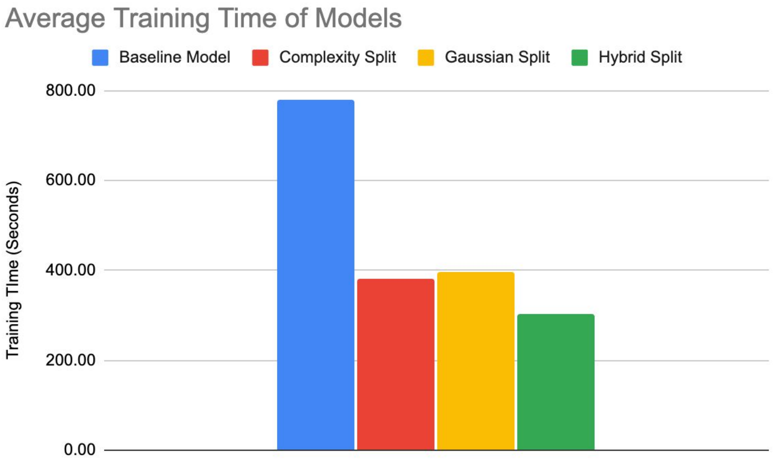
# Results and Comparison

Pros:

- Large decrease in training time
- Comparable results to base models
- In some strategies splitting the dataset may be very easy, only benefitting the model.

Cons:

- If used on large pre-existing datasets some strategies may take a lot of time and manual labor
- On some datasets there might be catastrophic forgetting if we don't revisit previous 'difficulties'



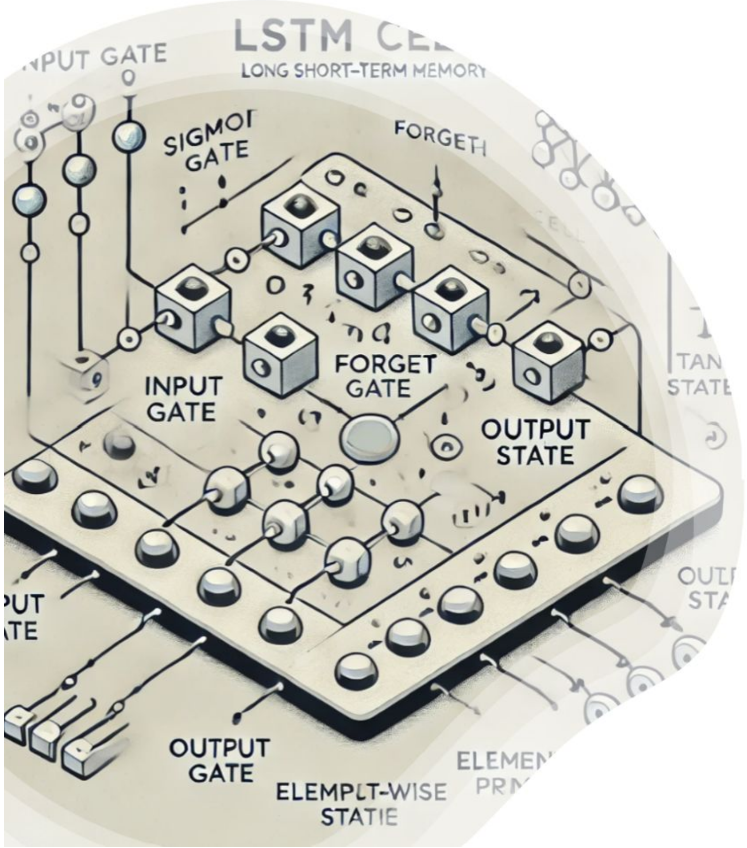

| Column 1 | Baseline Model | Model 1 (using Complexity Split) | Model 2 (using Gaussian Split) | Model 3 (using Hybrid) |
|---|---|---|---|---|
| Accuracy on Normal Data | | | | |
| Run 1 | 97.53 | 97.78 | 96.89 | 97.04 |
| Run 2 | 97.47 | 97.12 | 97.19 | 97.33 |
| Run 3 | 97.68 | 97.68 | 97.14 | 97.53 |
| Average Percentage | 97.56 | 97.53 | 97.07 | 97.30 |
| | | | | |
| Accuracy on Gaussian Blur Testing Set | | | | |
| Run 1 | 96.79 | 97.63 | 97.19 | 96.99 |
| Run 2 | 97.09 | 96.64 | 97.58 | 96.89 |
| Run 3 | 96.64 | 96.3 | 96.79 | 97.73 |
| Average Percentage | 96.84 | 96.86 | 97.19 | 97.20 |

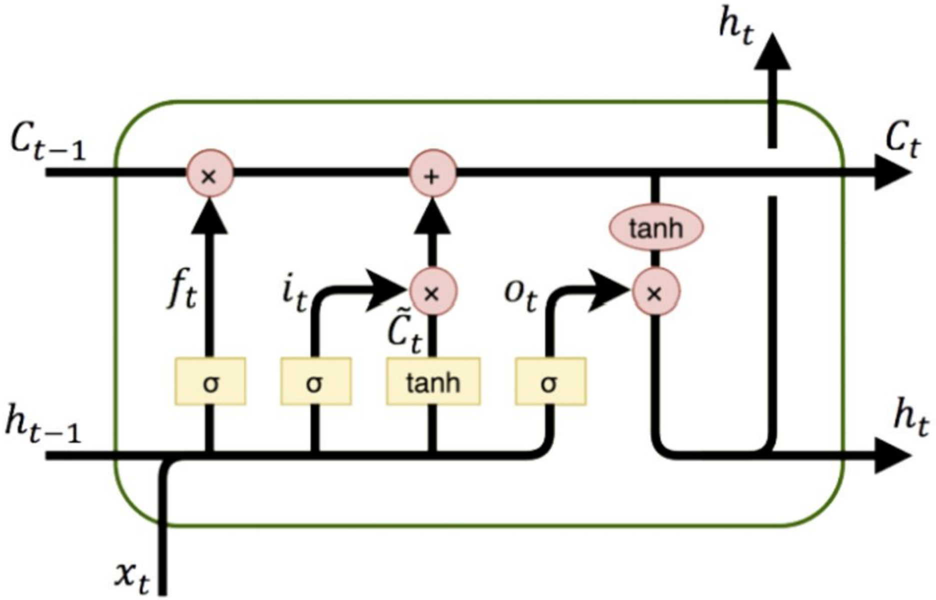
Similar results to previous papers, especially with Gaussian Blur models.

# Advancing TDDFT simulations using LSTM

Dongming Li (dongmingli@umass.edu)

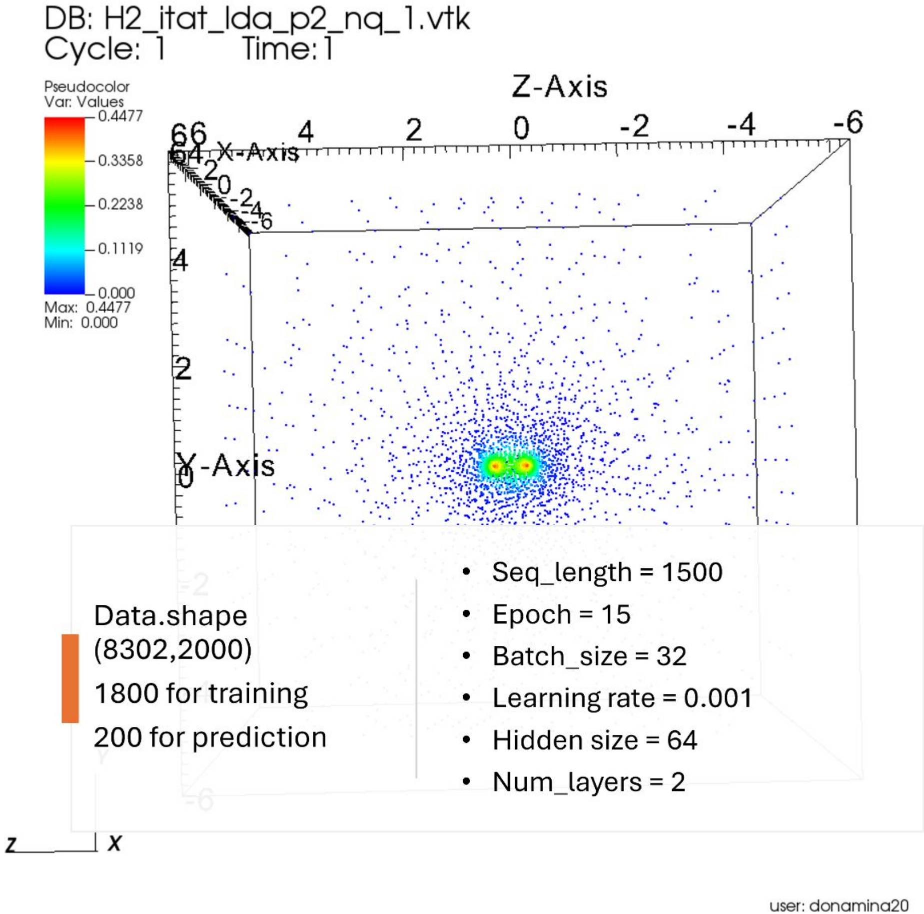
Project #38




# Long Short-Term Memory (LSTM) architecture

# Baseline and Results

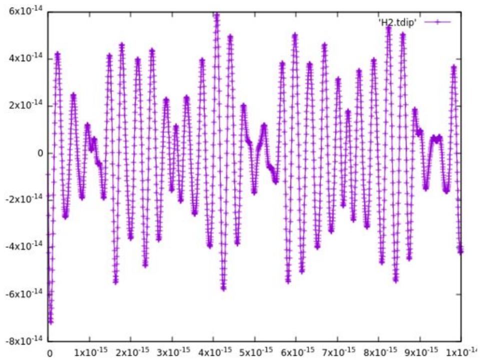


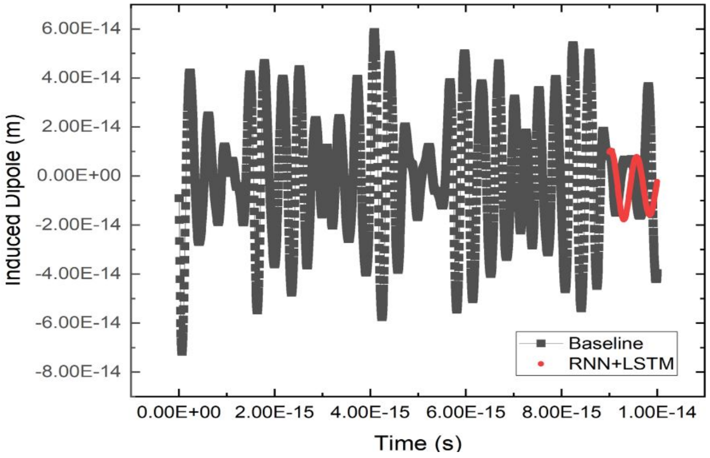

Data.shape (8302,2000)

1800 for training

200 for prediction

- Seq_length = 1500
- Epoch = 15
- Batch_size = 32
- Learning rate = 0.001
- Hidden size = 64
- Num_layers = 2



Machine-learning Kohn–Sham potential from dynamics in time-dependent Kohn–Sham systems

*Jun Yang, and James Whitfield*

*Machine learning Science and Technology, 2023*

## 4. Conclusions

In this article, we have described a new machine-learning method to learn the dynamics and the Kohn–Sham potential (or the hardest part—xc potential) of the Kohn–Sham system. We have demonstrated the method with two one-dimensional examples: a harmonic oscillator model and a two-electron soft Coulomb model. In both examples, the exact dynamics of the systems could be well reproduced from the machine learning method. The machine-learned potential in the harmonic oscillator test captures the general feature of the actual quadratic form potential, but it shows a discrepancy from the actual one in the two-electron test. We have analyzed the possible reasons, and the memory effect is the major source of error. The memory effect requires considering the densities of previous timestamps. To overcome this difficulty, we believe the neural networks capable of handling time series (e.g. RNN, LSTM) are promising [34, 36].

# Compact diffusion model for Cifar-10

Chuchen Li (chuchenli@umass.edu),
**Isaac Zhong (bzhong@umass.edu)**

Project #48

# Summary

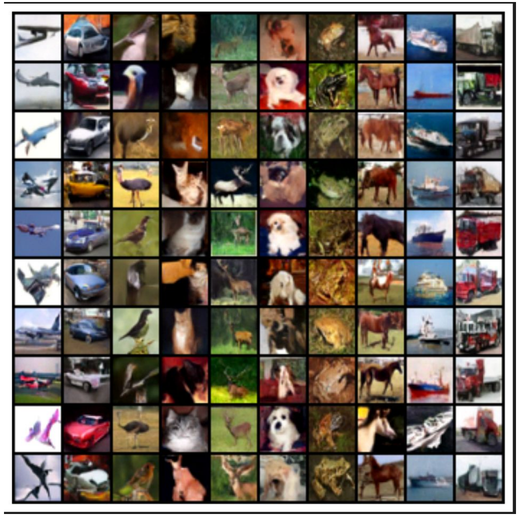
- **Motivation & Goal:** Reduce the size and inference time (time used to generate image) of the diffusion model while preserving the quality of the images generated by the model.
- **Dataset:** CIFAR-10 3x32x32
- **Background:**
  - We measure performance by Fréchet inception distance (FID) score (measures difference between images, the lower the better), inference time, and model size
  - We use Denoising Diffusion Probabilistic Model (DDPM), which is a kind of generative diffusion model. Generative diffusion model is trained by adding noise to train image and denoising. During inference, it operates the learned denoising process on pure noise to generate new images.
  - Conditional diffusion model improves training of model (compared to unconditional). Conditional model takes into account the label instead of purely depending on images in training data.
  - Exponential moving average (EMA) improves training of model. EMA smooth model's weight over time and thus enhance the model's stability when encountering fluctuation in training data.
  - Structural pruning to optimize model size and training time. Structural pruning removes redundant parameters from model by modifying its structure
  - Quantization optimize model size and training time. Quantization optimizes the model by reducing bit size of parameters (for example float 32 to int 8)

# Result

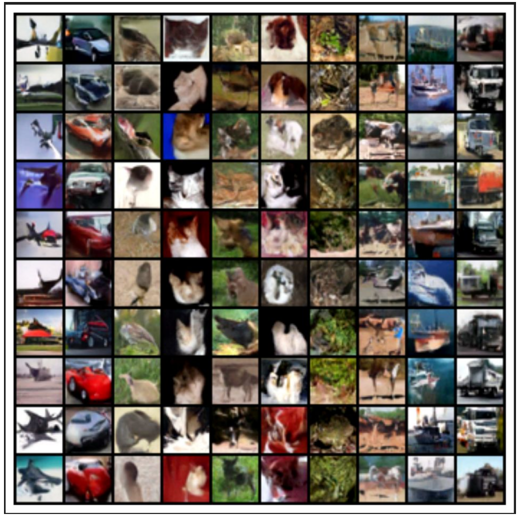
- We implemented an unconditional Denoising Diffusion Probabilistic Model (DDPM) as our base model
- We then implemented a conditional version of DDPM
- We added exponential moving average (EMA) on top of conditional DDPM
- We then implemented structural pruning on top of on top of conditional DDPM with EMA
- We at last added quantization, but our method is naive and only support CPU inference
- All inference except CPU inference are performed on 4080 laptop GPU

| | FID-(5000 images) | Model-Size (Mb) | Inference-time-10-images (seconds) |
|---|---|---|---|
| Unconditional | 46.6382 | 117.8429 | 22 |
| Conditional | 40.1885 | 23.6553 | 19 |
| Conditional-EMA | 35.2616 | 23.6553 | 19 |
| Conditional-EMA-Pruned | 46.2591 | 9.9578 | 19 |
| Conditional-EMA-Quantized | 48.9244 | 23.0638 | 150 (CPU) |
| Conditional-EMA-Pruned-Quantized | 58.8651 | 9.3689 | 133 (CPU) |

**Inference Result of Conditional-EMA-DDPM**



**Inference Result of Conditional-EMA-Pruned-DDPM**



# Conclusion

- Comparing between our results, The conditional-EMA implementation provides the best result with small enough model size, which is our recommended model for the CIFAR-10 dataset. Structural pruning may be preferred if smaller model size is required with tolerable increase in FID score.
- Dynamic quantization is not recommended as it provides too little decrease in model size compared to pruning but resulted in larger increase in FID score. Also it only supports CPU inference leading to unacceptably high inference time.

# Future Work

- Utilize another quantization method, which could possibly provide less loss in FID score
- Implement other strategies, like knowledge distillation, that can provide same picture quality while reduces size and inference time

# Classroom Learning with Knowledge Distillation

Matthew Peters (matthewpeter@umass.edu),
Neeladri Bhuiya (nbhuiya@umass.edu),
Shreyaa Dani (sdani@umass.edu)

Project #52

# Background and Motivation

**Project Objective**:

The goal is to develop a more efficient image classification system by applying **Knowledge Distillation** (KD) to an ensemble of student models. This allows us to replicate the performance of larger, deeper models while reducing computational costs

**Motivation**:

- **High Computational Costs of Large Models**: Models like ResNet110 and VGG-16 require substantial resources, making them impractical for deployment in resource-constrained environments (e.g., mobile devices).
- **Need for Efficient Models**: **Knowledge Distillation** and **ensemble methods** help improve smaller models' performance, making them viable for low-power devices without sacrificing accuracy

**Background:**

- **Ensemble Learning**: Combines predictions from multiple models to improve performance. By using multiple student models in an ensemble, performance can be maintained while significantly reducing computational complexity
- **Mixture of experts:** Uses a gating network to divert the processed inputs to the necessary expert/experts, to create a sparse model unlike the dense models that are traditionally used.
- **Knowledge Distillation:** Knowledge distillation (KD) is a technique where a smaller "student" model learns from a larger "teacher" model, transferring knowledge through softened output probabilities. In our code, KD is implemented by using the teacher model's logits (raw predictions) as soft targets for the student model, which helps the student model capture finer details of the data distribution
- Combined **loss function** used in training the student consists of the standard **cross-entropy loss** (to match hard labels) and a **KL divergence** loss (to align the student's predictions with the teacher's softened outputs)
    - $\text{Loss} = \alpha \cdot \text{Cross-Entropy Loss} + \beta \cdot \text{KL Divergence}$
    - Where $\alpha$ and $\beta$ are weights that control the importance of each component of the loss.

# Results (Ensemble)

| Teacher Models | | |
|---|---|---|
| **Model** | **Accuracy (%)** | **Parameters (M)** |
| ResNet32 | 81.82 | 0.49 |
| ResNet56 | 86.21 | 0.88 |
| ResNet110 | 87.37 | 1.75 |

* CNNs have 2 convolutions (3x3 Kernel, stride 1, padding 1, 16 filters) each

| Student Model | Teacher Model | Number of Students | Accuracy before Dist. (%) | Accuracy after Dist. (%) | Improvement (%) | Compression Ratio |
|---|---|---|---|---|---|---|
| CNN* | ResNet32 | 1 | 42.76 | 43.81 | +1.05 | 0.0062 |
| CNN* | ResNet32 | 5 | 46.16 | 47.50 | +1.34 | 0.0308 |
| CNN* | ResNet32 | 8 | 47.31 | 47.92 | +0.61 | 0.0493 |
| CNN* | ResNet56 | 1 | 44.37 | 44.76 | +0.39 | 0.0034 |
| CNN* | ResNet56 | 5 | 43.47 | 45.13 | +1.66 | 0.0171 |
| CNN* | ResNet56 | 10 | 45.71 | 47.71 | +2.00 | 0.0343 |
| ResNet8 | ResNet56 | 1 | 69.18 | 80.10 | +10.92 | 0.1124 |
| ResNet8 | ResNet56 | 5 | 77.49 | 84.01 | +6.52 | 0.5622 |
| ResNet8 | ResNet56 | 9 | 78.32 | 85.36 | +7.04 | 1.0119 |
| ResNet14 | ResNet56 | 1 | 74.25 | 84.51 | +10.26 | 0.2234 |
| ResNet14 | ResNet56 | 3 | 77.44 | 87.06 | +9.62 | 0.6701 |
| ResNet14 | ResNet56 | 5 | 81.57 | 87.53 | +5.96 | 1.1169 |

# Results (Mixture of Experts)

| Student Model | Teacher Model | Number of experts | Accuracy before Dist. (%) | Accuracy after Dist. (%) | Improvement (%) | Effective Compression Ratio |
|---|---|---|---|---|---|---|
| MoE | ResNet32 | 2 | 65.09 | 63.87 | -1.22 | 0.097 |
| MoE | ResNet32 | 3 | 64.99 | 64.51 | -0.48 | 0.097 |
| MoE | ResNet32 | 4 | 65.47 | 63.87 | -1.6 | 0.097 |
| MoE | ResNet32 | 5 | 66.29 | 66.07 | -0.22 | 0.097 |
| MoE | ResNet32 | 6 | 65.73 | 64.34 | -1.39 | 0.097 |
| MoE | ResNet32 | 7 | 64.99 | 65.84 | +0.85 | 0.097 |
| MoE | ResNet32 | 8 | 64.63 | 64.57 | -0.06 | 0.097 |
| MoE | ResNet32 | 9 | 64.92 | 65.86 | +0.94 | 0.097 |
| MoE | ResNet32 | 10 | 64.66 | 65.21 | +0.55 | 0.097 |

# Conclusion and key takeaways

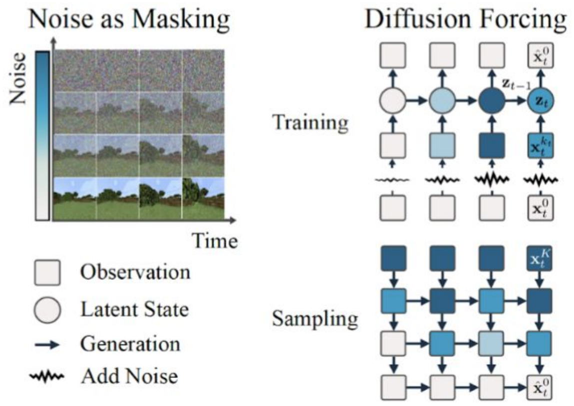
- Ensembles of CNN and ResNet student models achieved actuaries near to their teacher models
- Simple CNN student achieved compression ratio as low as 0.0062 with respectable accuracy upto 47% with Resnet32 as teacher.
- Resnet8 student ensemble trained w Resnet56 achieved 84.01% accuracy i,e 98% of teacher's performance while having half parameter count.
- Students trained via distillation outperformed those trained independently by nearly 2% in CNN and 6-10% in Resnet8 models.
- Gains are the highest when the teacher and student have a similar architecture.
- Mixture of experts students don't work well with dense model teachers due to disparity in the loss objectives.
- ResNets outperform MoE models with similar parameter size.

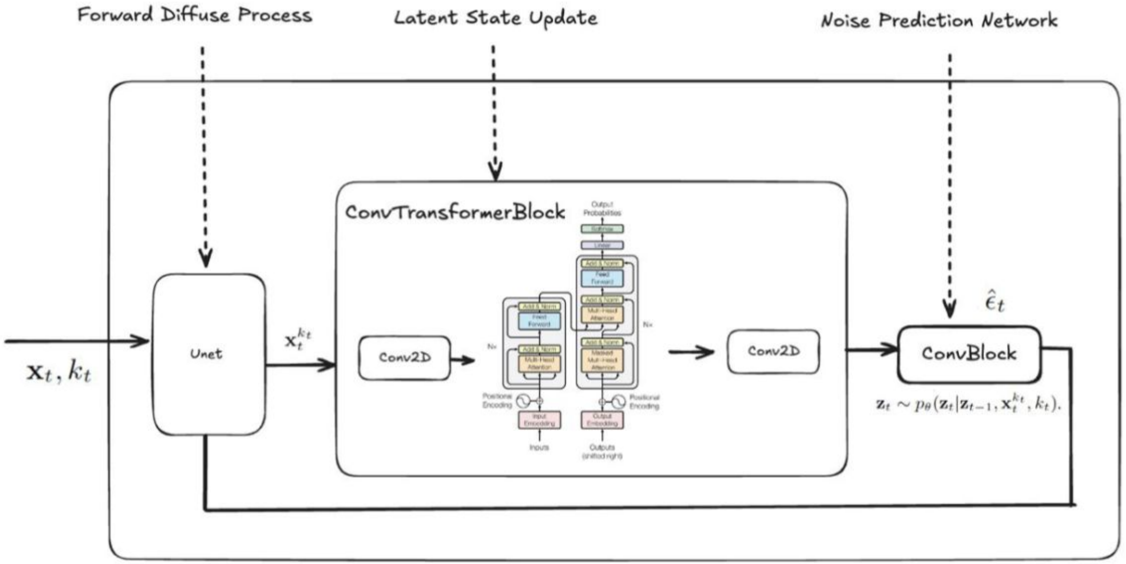
# Investigating Diffusion Forcing Algorithm for Long-Horizon Robot Imitation Learning

Anushka Agarwal (anushkaagarw@umass.edu),
Muhammad Yusuf Hassan (mdhassan@umass.edu),
**Talha Chafekar (tchafekar@umass.edu)**

Project #53

# Motivation and Background

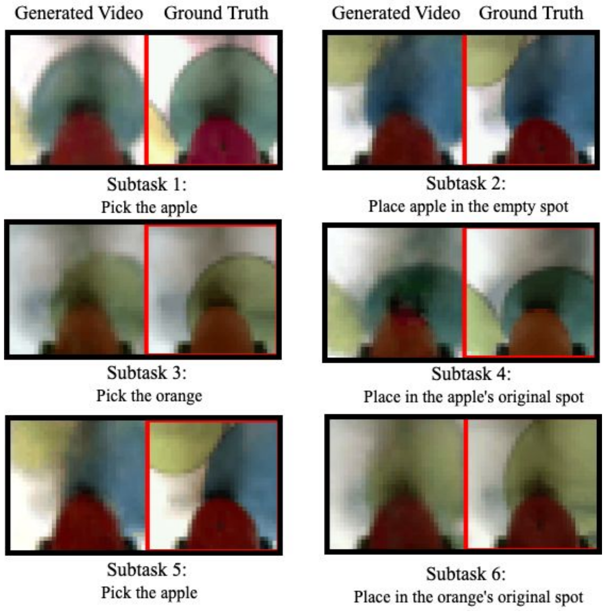


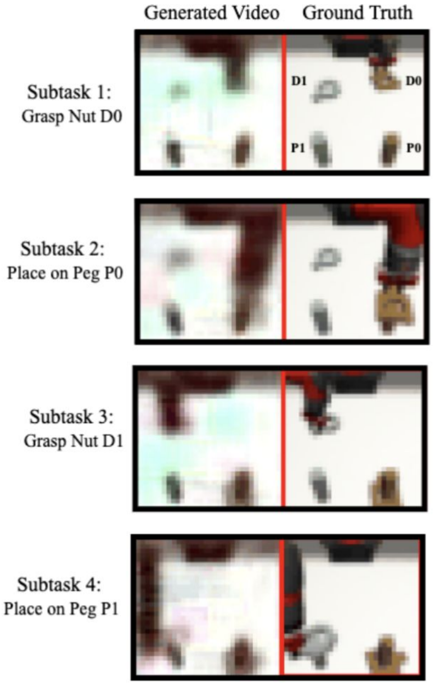




## Why Diffusion Forcing?

- Independent noise levels for each time step, making it robust.
- Variable length sequence generation potentially helpful for long horizon robotics tasks.

# Main Results



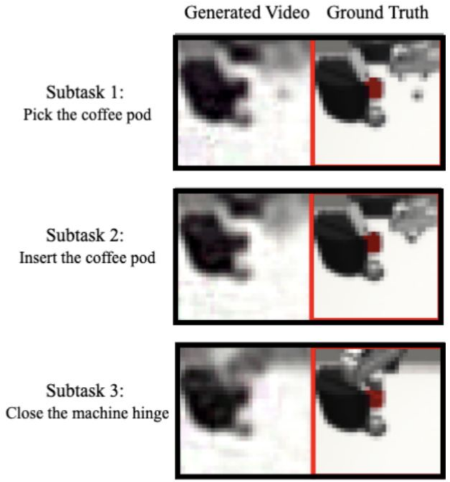




Table 3. Comparison of results for the Coffee task

| Method | MSE ↓ | UIQI ↑ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|---|
| Diffusion Forcing (RNN) | 0.452 | 0.119 | 0.258 | 9.46 |
| Diffusion Forcing (Transformer) | **0.445** | **0.127** | **0.268** | **9.53** |

Table 1. Comparison of results for the Fruit Swap task

| Method | MSE ↓ | UIQI ↑ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|---|
| Diffusion Forcing (RNN) | 0.092 | **0.414** | **0.501** | 16.390 |
| Diffusion Forcing (Transformer) | **0.090** | 0.412 | **0.501** | **16.464** |

Table 2. Comparison of results for the Nut Assembly task

| Method | MSE ↓ | UIQI ↑ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|---|
| Diffusion Forcing (RNN) | 0.317 | **0.189** | 0.179 | 11.00 |
| Diffusion Forcing (Transformer) | **0.296** | 0.180 | **0.264** | **11.296** |

# Conclusion

- Our proposed Transformer network performs better for multiple long horizon tasks as compared to the baseline RNN.
- Diffusion Forcing is intensive with respect to resources.
- Diffusion Forcing is quite sensitive to hyperparameter tuning.

# Knowledge distillation for efficient neural network compression

Akshata Sangwai (asangwai@umass.edu),
**Prateek Bhindwar (pbhindwar@umass.edu)**

Project #55

# Goal

Create a custom model that surpasses the baseline accuracy of existing designs by combining various established methods and architectures within the knowledge distillation framework while significantly compressing the teacher model.

# Motivation

Knowledge distillation is traditionally achieved using KL-Divergence loss between the teacher's outputs and the hard labels. However, since the teacher is a large and complex network, the student's architecture is often constrained in terms of the number of layers, as excessive compression can lead to information loss.

# Background

We experimented with the following methods for calculating loss:

1. Intermediate feature based distillation
2. Mutual loss based distillation

In order to design the custom lite model we studied the following architectures:

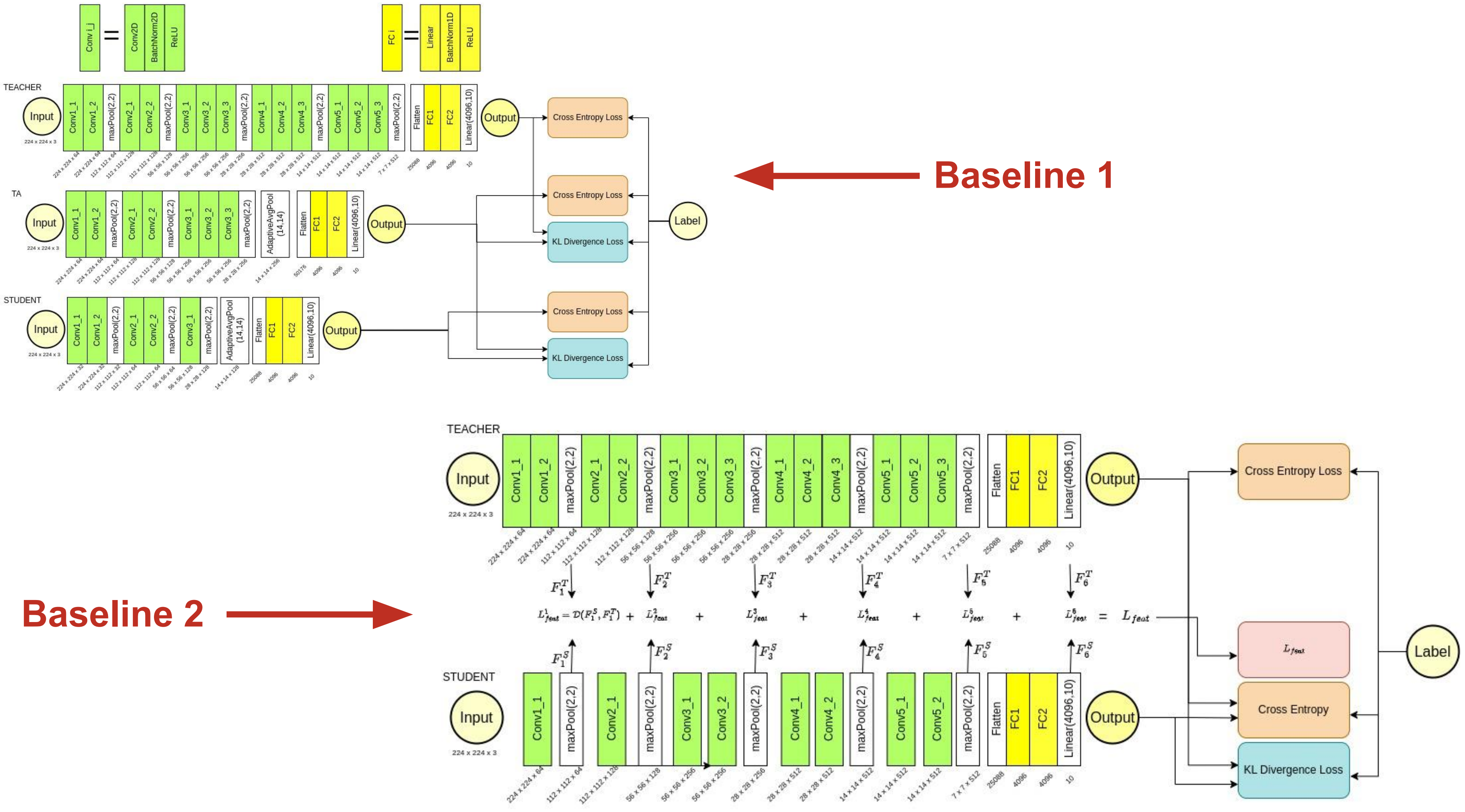
1. TAKD - Teacher Assistant Knowledge Distillation
2. CMTKD - Collaborative Multi Teacher Knowledge Distillation

**Teacher Model:** VGG-16

**Dataset:** Imagenette (10 classes & 13,394 images)

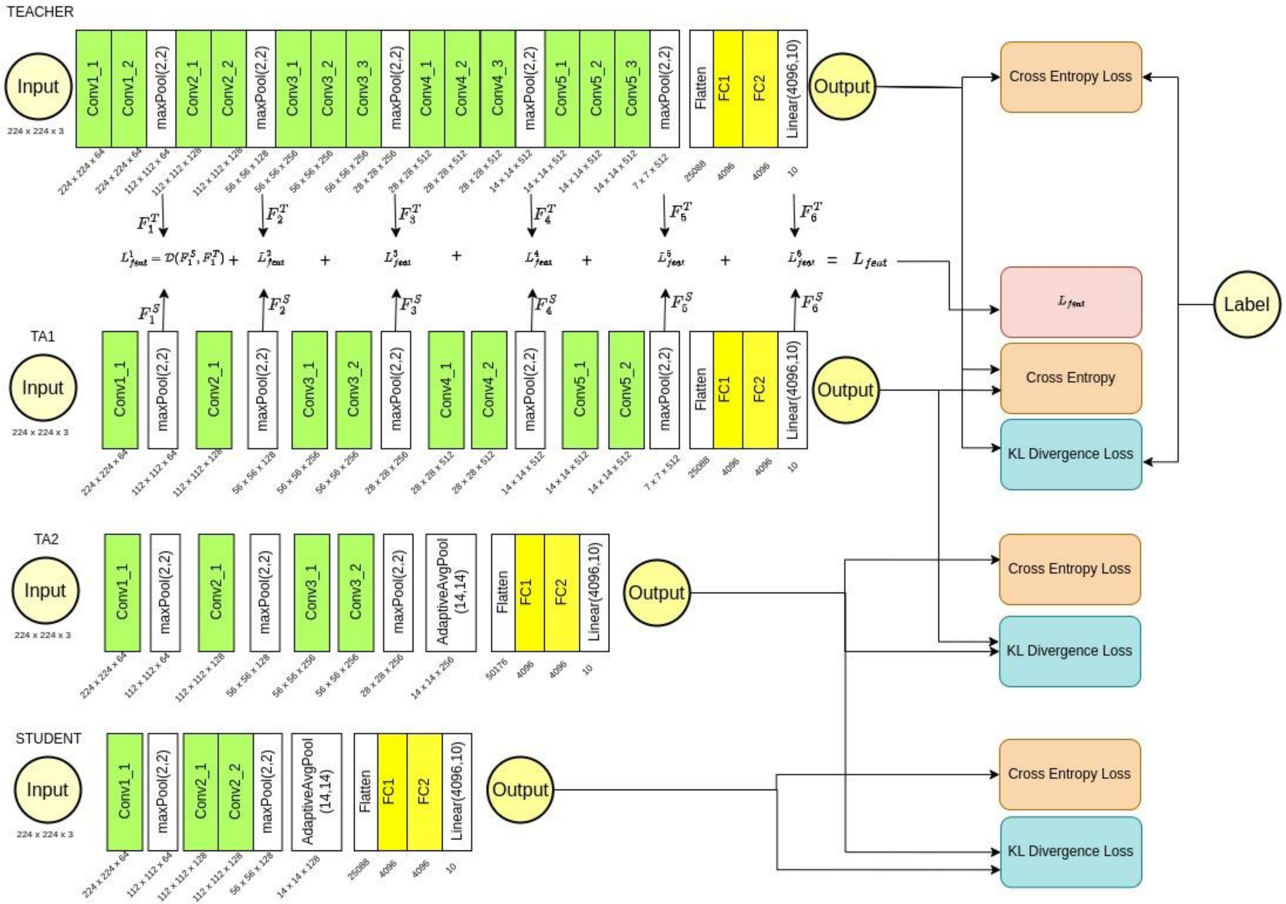
**References:**


1. TAKD paper
2. CMTKD paper



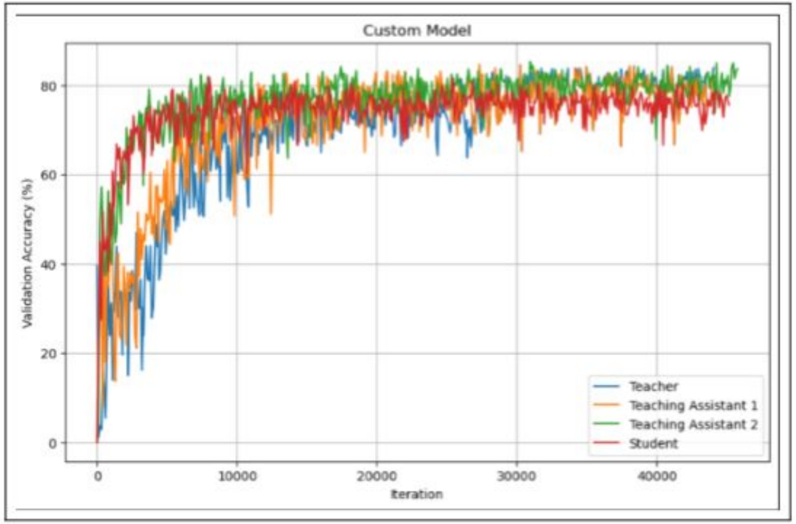
Conv i_j = Conv2D BatchNorm2D ReLU
FC i = Linear BatchNorm1D ReLU
TEACHER
Input
224 x 224 x 3
Conv1_1 Conv1_2 maxPool(2,2) Conv2_1 Conv2_2 maxPool(2,2) Conv3_1 Conv3_2 Conv3_3 maxPool(2,2) Conv4_1 Conv4_2 Conv4_3 maxPool(2,2) Conv5_1 Conv5_2 Conv5_3 maxPool(2,2)
Flatten FC1 FC2 Linear(4096,10)
Output
TA
AdaptiveAvgPool (14,14)
STUDENT
Cross Entropy Loss
KL Divergence Loss
Label
Baseline 1
Baseline 2
$L^1_{feat} = \mathcal{D}(F^S_1, F^T_1)$ + $L^2_{feat}$ + $L^3_{feat}$ + $L^4_{feat}$ + $L^5_{feat}$ + $L^6_{feat}$ = $L_{feat}$
Cross Entropy

# Custom Lite Architecture



## Results

- ➔ The graph on the right shows validation accuracy vs iterations for the Custom Lite architecture.
- ➔ Table 1 compares the compression ratio and performance gain of all three architectures.
- ➔ Table 2 compares the validation and test accuracies of all the three architectures




| Method | Compression Ratio | Performance Gain |
|---|---|---|
| Baseline 1 | 2.6 | 1.33% |
| Baseline 2 | 1.625 | 1.16% |
| Custom Lite | 4.334 | 0.92% |

| Method | Model | Validation Accuracy | Test Accuracy | Number of CNN Blocks |
|---|---|---|---|---|
| Baseline 1 | Teacher | 83.90 | 66.40 | 13 |
| Baseline 1 | Teaching Assistant | 86.60 | 73.40 | 7 |
| Baseline 1 | Student | 84.60 | 67.62 | 5 |
| Baseline 2 | Teacher | 83.90 | 66.40 | 13 |
| Baseline 2 | Student | 84.70 | 67.45 | 8 |
| Custom Architecture | Teacher | 83.90 | 66.40 | 13 |
| Custom Architecture | Teaching Assistant 1 | 84.70 | 67.45 | 8 |
| Custom Architecture | Teaching Assistant 2 | 85.30 | 69.64 | 4 |
| Custom Architecture | Student | 81.80 | 67.21 | 3 |

# Conclusion

1. We were able to successfully implement Baseline 1 and Baseline 2 after a few tweaks in their original design, given our computational constraints.

2. We successfully created a custom architecture (Custom Lite) which incorporates both the methods of Knowledge Distillation - Intermediate feature-based KD and Mutual Loss-based KD

3. In our Custom Lite architecture, we achieved a compression ratio of 4.334 and our lightweight student still managed to surpass its teacher's test accuracy (67.21% compared to 66.40%)

4. The most important part of KD is the KL-Divergence loss. In both methods of KD, the KL-Divergence between output distribution of the teacher and student plays a critical role in guiding the student model to mimic the teacher's behavior.

5. This loss function measures how well the student model's predictions align with the softened probability distribution of the teacher, which often contains richer information about class relationships than the hard labels.

6. (**Future Scope:** ) Incorporating either multiple teachers with varying quantization levels or multiple teachers with different architectures while maintaining the same number of modules (e.g., ResNet).

# Developing A Compact Diffusion Model

Ajit Kumar (ajitkumar@umass.edu),
Pronay Dutta (pronaydutta@umass.edu)

Project #60

# Introduction & Objectives

- **Diffusion Models**:
  - Powerful generative models for high-fidelity image generation.
  - Operate through a gradual denoising process.
- **Challenges**:
  - Computationally intensive and memory-heavy.
  - Difficult deployment on resource-constrained devices.
- **Objectives**:
  - Develop a compact diffusion model maintaining high image quality.
  - Apply model compression techniques:
    - **Knowledge Distillation**: Transfer knowledge from a larger teacher to a smaller student model.
    - **Pruning**: Eliminate redundant parameters.
    - **Quantization**: Reduce numerical precision for further optimization.
  - Evaluate performance based on image quality, inference time, and model size.

# Architecture & Methodology

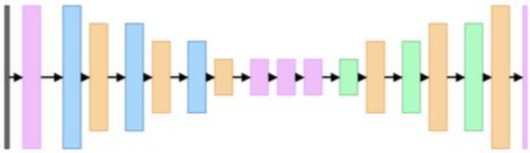
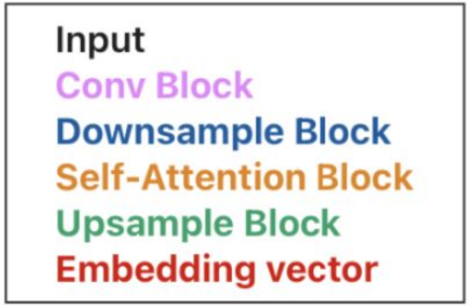


Figure: U-Net Architecture Overview




U-Net backbone with self-attention and convolutional layers to capture local and global features.

**Methodology:**

- **Knowledge Distillation:** Combines standard diffusion loss ($L_{\text{standard}}$) and distillation loss ($L_{\text{distill}}$):

$$L_{\text{student}} = \alpha L_{\text{standard}} + \beta L_{\text{distill}}$$

- **Mixed Precision Training and EMA:** Mixed precision combines FP16/FP32 to reduce memory usage, while Exponential Moving Average (EMA) smooths weight updates for stability.
- **Pruning:** Magnitude-based channel pruning removes less significant channels based on their L1 norm. Fine-tuning ensures performance is maintained after pruning.

# Results & Future Directions

**Performance Metrics**

| Model | Size (MB) | Inference Time (s) | FID Score |
|---|---|---|---|
| Teacher Model | 89.02 | 18.54 | 163 |
| Student Model | 22.6 | 15.72 | 184 |
| Pruned Student Model | 17.42 | 15.4 | 213 |

**Future Directions**

- Implement more efficient training techniques.
- Increase the number of training images for better generalization.
- Explore advanced pruning and quantization methods.
- Utilize symmetries and latent space diffusion models.
- PHYSICS: RG flow vs diffusion models?

# COMEDI: Under Controlled Image Emotion Editor

Gehao Zhang (gehaozhang@umass.edu),
Zekai Zhang (zekzhang@umass.edu)

Project #62

# Motivation and Background

- What is AIM?
  - Affective Image Manipulation.
  - Modifying user-provided images to evoke specific emotional responses.
- Motivations
  - Previous works offered uncontrolled editing, where users provided an image and target emotion but had no control over the outcome, often leading to edits that conflicted with users intent.
  - Existing editors either require large-scale training or rely on calling proprietary LLMs
- Our Goals
  - The first image emotion editor to evoke specific emotions while integrating user instructions for controlled edits.
  - The first local deployable emotion editor.

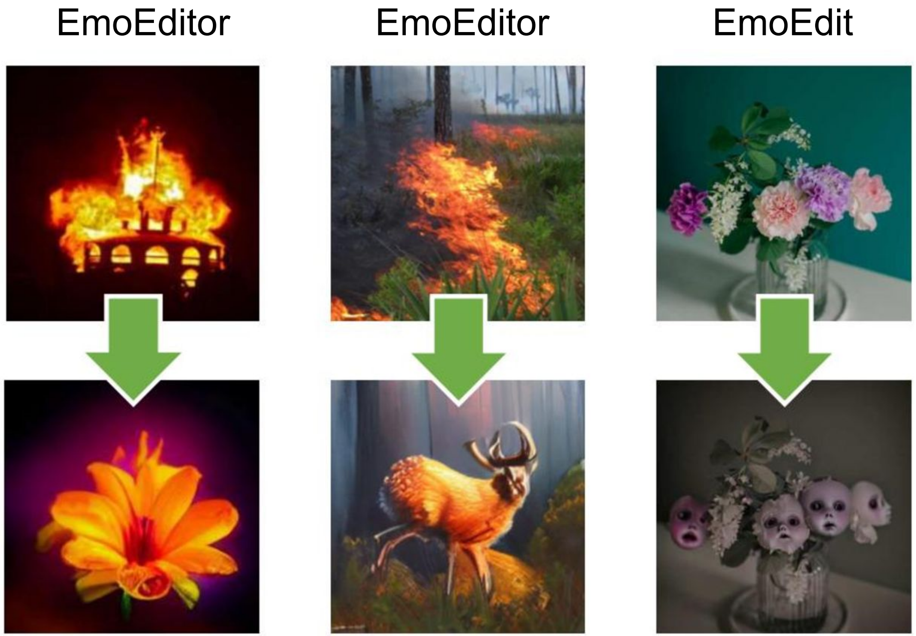



Figure 1. Editing results from previous works.

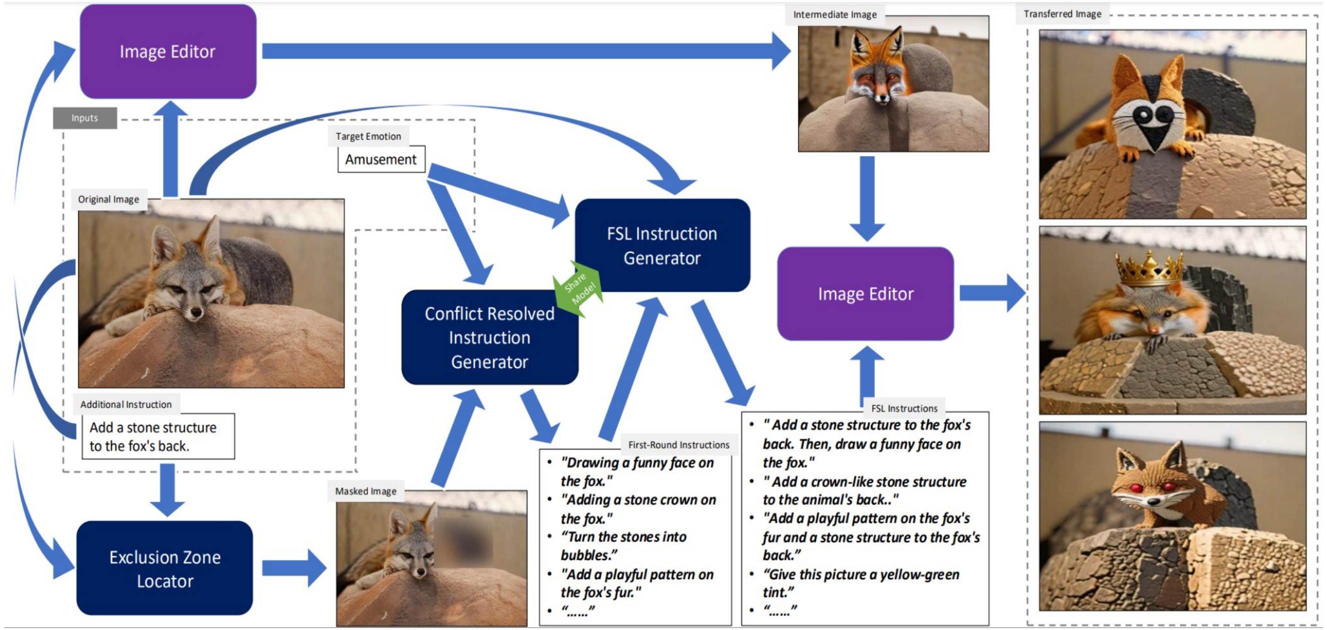
# CoMEdi: A multi-step agent framework

Step 1: Generate an intermediate image based on user instruction by InstructPix2Pix

Step 2: Use LISA to blur specific area of original image that mentioned in user instruction

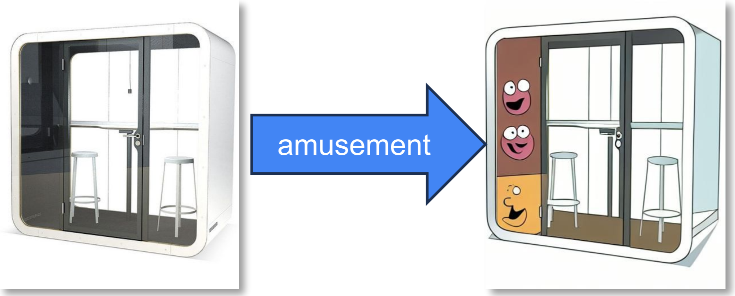
Step 3: Use LLAVA to generate emotional editing instructions from blurred image

Step 4: Feed emotional editing instructions and intermediate image to editor and get final results



# Results and Conclusion

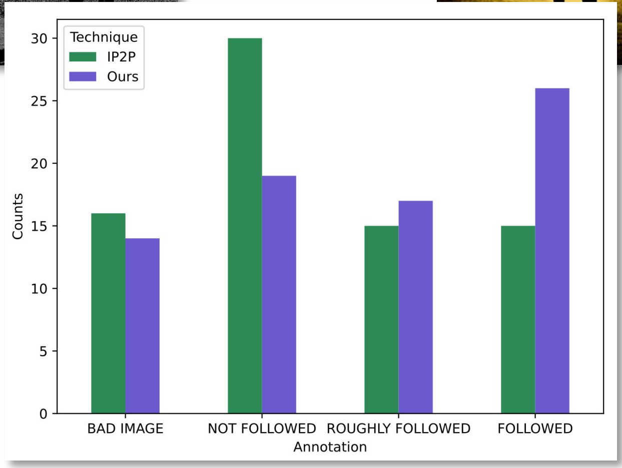
- 1.88× of emotion-manipulating ability compared to the best-performed baseline.
- 43% increase in additional instruction following ability





| Method | Metric | Positive Emotions | | | | Negative Emotions | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| | | amusement | awe | contentment | excitement | anger | disgust | fear | sadness | |
| UltraEdit [4] | $\Delta^E$ | 0.126 | 0.013 | 0.040 | 0.132 | 0.044 | 0.031 | 0.030 | -0.013 | 0.051 |
| IP2P w/ MagicBrush [5] | | 0.115 | 0.038 | 0.048 | 0.128 | 0.055 | 0.024 | 0.052 | 0.020 | 0.060 |
| InstructPix2Pix (IP2P) [3] | | 0.147 | 0.087 | 0.114 | 0.135 | 0.101 | 0.047 | 0.099 | 0.058 | 0.098 |
| COMEDI (Ours) | | **0.270** | **0.157** | **0.185** | **0.215** | **0.146** | **0.051** | **0.304** | **0.141** | **0.184** |
| UltraEdit | $\Delta^E_{SSIM}$ | 0.082 | -0.001 | 0.019 | 0.098 | 0.035 | 0.012 | 0.014 | -0.011 | 0.031 |
| IP2P w/ MagicBrush | | 0.057 | 0.005 | 0.017 | 0.073 | 0.039 | 0.014 | 0.026 | 0.009 | 0.030 |
| InstructPix2Pix (IP2P) | | 0.070 | 0.024 | 0.044 | 0.076 | 0.056 | **0.020** | 0.046 | 0.028 | 0.045 |
| COMEDI (Ours) | | **0.128** | **0.054** | **0.074** | **0.117** | **0.062** | 0.020 | **0.117** | **0.066** | **0.080** |
| UltraEdit | $\Delta^E_{CLIP}$ | 0.098 | 0.008 | 0.024 | 0.108 | 0.033 | 0.020 | 0.017 | -0.014 | 0.037 |
| IP2P w/ MagicBrush | | 0.090 | 0.018 | 0.030 | 0.106 | 0.048 | 0.021 | 0.040 | 0.013 | 0.046 |
| InstructPix2Pix (IP2P) | | 0.103 | 0.046 | 0.068 | 0.103 | 0.074 | 0.034 | 0.073 | 0.030 | 0.067 |
| COMEDI (Ours) | | **0.185** | **0.094** | **0.106** | **0.155** | **0.095** | **0.036** | **0.205** | **0.096** | **0.122** |

Table 2. Evaluation results among COMEDI and baselines



# Enhancing text classification with llm driven augmentation for imbalanced datasets

Chandana Pamidi (cpamidi@umass.edu),
Mahima Choudha (mchoudha@umass.edu),
**Ujwala Munigela (umunigela@umass.edu)**

Project #65

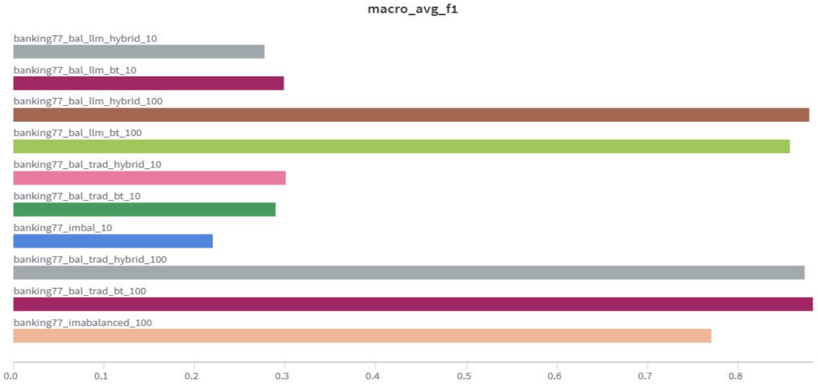
# Project Summary

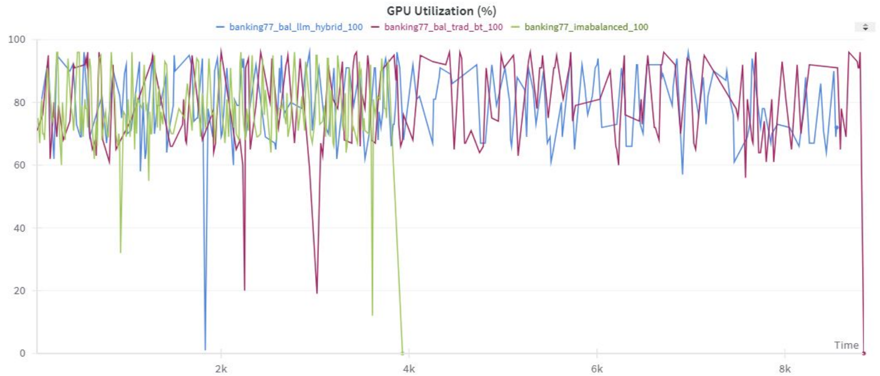
UMassAmherst | Manning College of Information & Computer Sciences

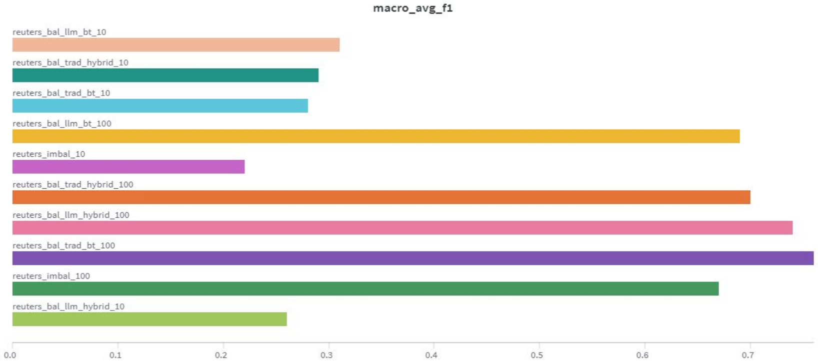
- Objective
  - Improve text classification performance by augmenting data for under-represented categories using LLMs.
- Datasets
  - Banking77: 77 classes, granular intent classification.
  - Reuters-21578: 8 classes, broad text categorization.
- Data Augmentation Strategies
  - Techniques: Cross-linguistic and Hybrid.
  - Approaches: Implemented using both Traditional (rule-based) and LLM (Gemini model).
- Classification Model
  - Mistral LLM is used for evaluating model performance.
- Goal
  - The goal is to evaluate whether data augmentation technique can improve accuracy for textual classification. Furthermore we are comparing different data augmentation techniques with traditional and LLM-based augmentation  and improve downstream classification results.

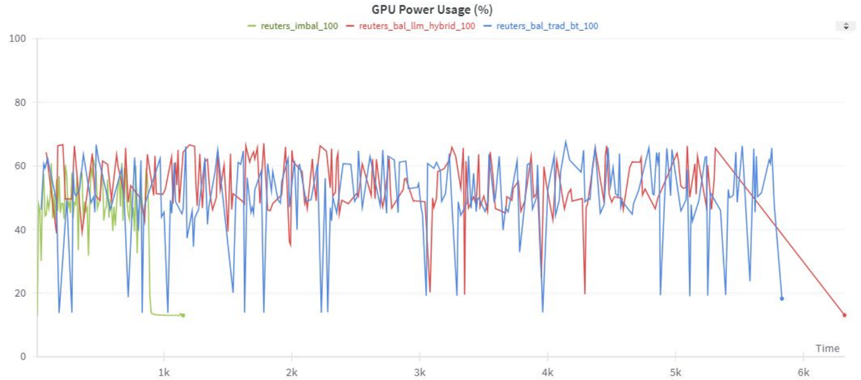
# Results - Performance and Cost

UMassAmherst | Manning College of Information & Computer Sciences



# Conclusion



- Balanced datasets demonstrate reduced bias compared to imbalanced datasets, leading to more reliable classification outcomes.
- With seed 10 Traditional approach is performing better compared to LLM Approach whereas for seed 100 LLM approach is performing better.

| | LLM Approach | Traditional Approach |
|---|---|---|
| seed = 10 | Cross Linguistic<br>banking77=32% reuters=30% | Hybrid<br>banking77=33% reuters=31% |
| seed = 100 | Hybrid<br>banking77=88% reuters=78% | Cross Linguistic<br>banking77=88% reuters=76% |

# Camera Tracking: A Human-Labeled View into Camera Dynamics

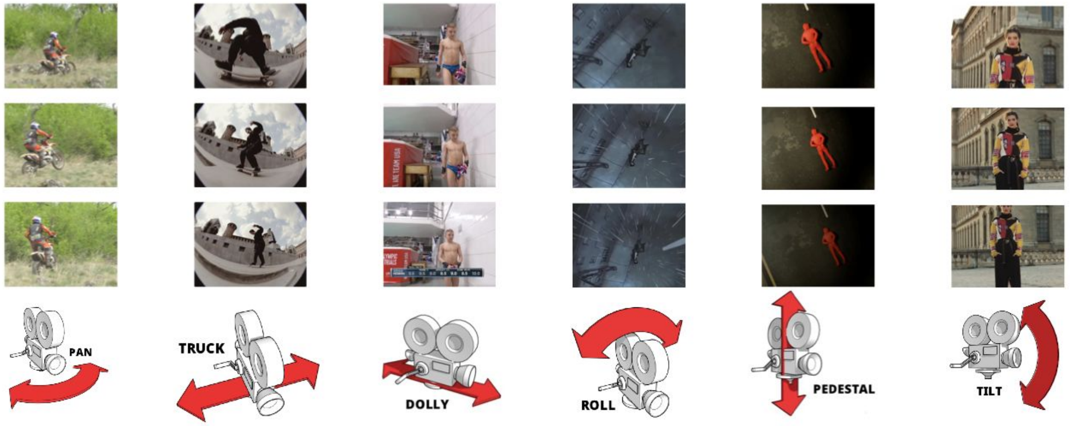
Delin Chen (delinchen@umass.edu),
Fengming Shen (fengmingshen@umass.edu),
**Siyuan Cen** (scen@umass.edu)

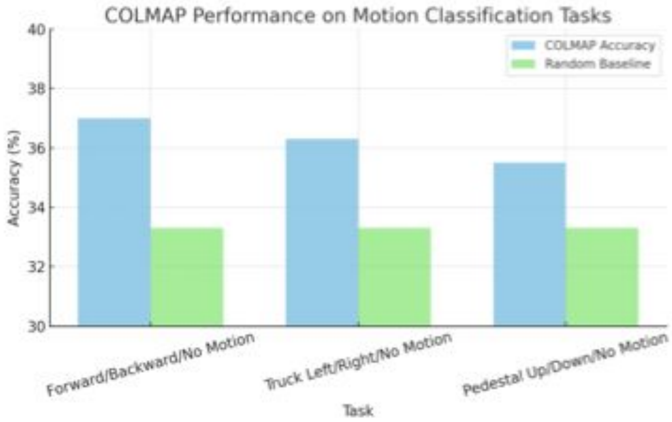
Project #66

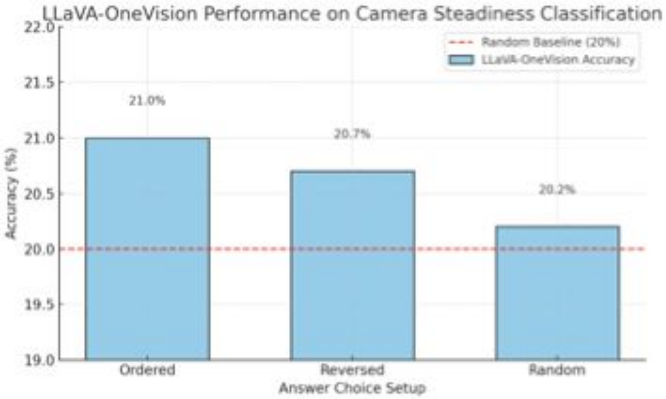
**Problem**: Existing machine learning models and tools, such as COLMAP and Vision-Language Models (VLMs), fail to accurately classify and interpret camera motion. Challenges include ambiguous motion cues, overlapping object and camera movements, and the absence of human-labeled datasets specifically tailored to camera dynamics.

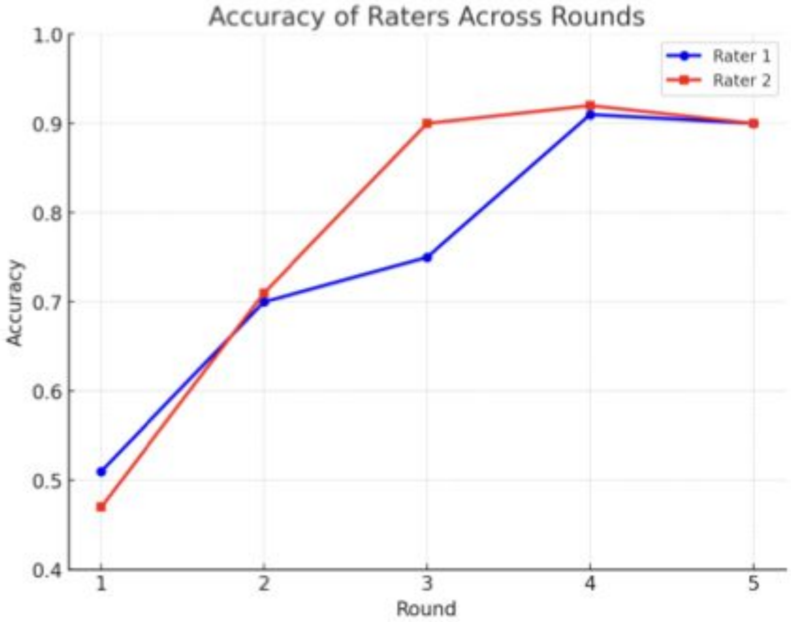
**Goal**: Develop a comprehensive human-labeled dataset and design a pseudo-automated labeling system to capture camera motion types and steadiness levels. This will enable more effective classification and interpretation of dynamic video content while improving scalability for future research.

**Approach**: Create a taxonomy of camera motion, label real-world video data with trained and professional human annotators, and evaluate the dataset using automated tools (e.g., COLMAP) and Vision-Language Models. Highlight the limitations of existing models while emphasizing the value of human annotations for improving motion-aware AI systems.




COLMAP Performance on Motion Classification Tasks
COLMAP Accuracy
Random Baseline
Accuracy (%)
40
38
36
34
32
30
Forward/Backward/No Motion
Truck Left/Right/No Motion
Pedestal Up/Down/No Motion
Task


LLaVA-OneVision Performance on Camera Steadiness Classification
Random Baseline (20%)
LLaVA-OneVision Accuracy
Accuracy (%)
22.0
21.5
21.0
20.5
20.0
19.5
19.0
21.0%
20.7%
20.2%
Ordered
Reversed
Random
Answer Choice Setup


Accuracy of Raters Across Rounds
Rater 1
Rater 2
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
1
2
3
4
5
Round

# Conclusion:

**Challenges in Camera Trajectory Understanding**
Camera motion classification is a complex problem, requiring nuanced temporal reasoning and the ability to disentangle overlapping object and camera dynamics.

**Dataset Contributions**
This work presents a human-labeled dataset designed to address these challenges, providing detailed annotations that set a new benchmark for motion analysis.

**Model Limitations**
State-of-the-art models like ”LLAVA-OneVision" and tools like COLMAP struggle to capture motion-specific cues, underscoring the need for motion-aware architectures.

# Overcoming the Animation Bottleneck: Evaluating Neural Style Transfer for Multi-Style Scalability in Creative Film Production

Benjamin Hall (bmhall@umass.edu),
**David Gerard (dgerard@umass.edu)**

Project #70

# Project Overview


**Motivation:** Style transfer as a tool to enhance animation film time and labor-intensive production, high computational resource cost, visual identity standardization, and multi-style project scalability.

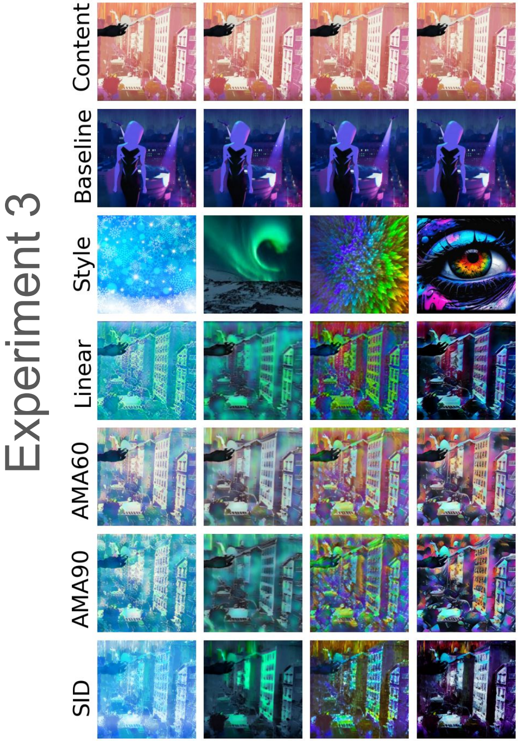
**Background:**

- Style Transfer Methods: Linear Style Transfer (Gram-Based), Arbitrary Style Transfer via Multi-Adaptation Network (Attention-Based) , Style Injection in Diffusion (Diffusion-Based)
- Evaluation Metrics: Art-Fid, structural similarity, color similarity, content loss, style loss, novel Animation Score, and inference time

**Summary:** Perform four style transformation experiments leveraging a designed scalable framework for the inference and evaluation pipeline.

# Experimental Results

**4 Primary Experiments:** Style transfer with comic images, style transfer with AI-generated images, emotion-based style transfer, and consecutive frames style transfer

# Conclusion

- Careful style image curation and trade-off prioritization → all models have high potential
- Diffusion technique highest quality stylized images at expense of inference efficiency, AMA60 best Animation Score
- Facilitates animation efficiency and creativity.

| **Model** | **ArtFID** | **SSIM** (avg ± std) | **ColorSim** (avg ± std) | **ContentLoss** (avg ± std) | **StyleLoss** (avg ± std) | **AnimationScore** | **AvgTime** (avg ± std) | **LoadTime** |
|---|---|---|---|---|---|---|---|---|
| AMA60 | 40.04 | 0.558 ± 0.146 | 0.147 ± 0.171 | 21066.62 ± 8491.96 | $1.1829 \times 10^{16} \pm 1.0174 \times 10^{16}$ | 2.89 | 0.166 ± 0.411 | 0.145 |
| Linear | 38.18 | 0.400 ± 0.149 | 0.290 ± 0.157 | 37077.58 ± 13970.67 | $1.9099 \times 10^{15} \pm 8.0571 \times 10^{14}$ | 1.34 | 0.510 ± 1.337 | 0.271 |
| AMA90 | 37.94 | 0.372 ± 0.134 | 0.324 ± 0.165 | 42009.41 ± 12775.43 | $5.5629 \times 10^{15} \pm 6.0418 \times 10^{15}$ | 2.81 | 0.175 ± 0.437 | 0.139 |
| SID | 44.90 | 0.569 ± 0.113 | 0.372 ± 0.187 | 19085.72 ± 5487.76 | $8.7546 \times 10^{15} \pm 4.5552 \times 10^{15}$ | 0.52 | 3.533 ± 0.014 | 20.860 |

Table 2. Evaluation Results for Comic Style Conversions

| **Model** | **ArtFID** | **SSIM** (avg ± std) | **ColorSim** (avg ± std) | **ContentLoss** (avg ± std) | **StyleLoss** (avg ± std) | **AnimationScore** | **AvgTime** (avg ± std) | **LoadTime** |
|---|---|---|---|---|---|---|---|---|
| AMA60 | 40.68 | 0.661 ± 0.058 | 0.009 ± 0.212 | 25613.93 ± 2261.33 | $9.1555 \times 10^{16} \pm 9.8551 \times 10^{16}$ | 1.92 | 0.342 ± 0.575 | 0.153 |
| Linear | 47.29 | 0.436 ± 0.119 | 0.236 ± 0.306 | 54126.31 ± 7073.21 | $7.5489 \times 10^{16} \pm 4.8070 \times 10^{16}$ | 1.07 | 0.984 ± 1.694 | 0.255 |
| AMA90 | 43.03 | 0.481 ± 0.075 | 0.121 ± 0.207 | 50527.12 ± 9301.46 | $7.2467 \times 10^{16} \pm 6.5663 \times 10^{16}$ | 1.92 | 0.341 ± 0.573 | 0.146 |
| SID | 46.48 | 0.457 ± 0.145 | 0.535 ± 0.161 | 41936.68 ± 5289.11 | $2.1538 \times 10^{16} \pm 2.1021 \times 10^{16}$ | 0.51 | 3.543 ± 0.011 | 19.767 |

Table 4. Evaluation Results for Emotion Style Conversions

# Textual augmentation for medical transcriptions

**VAISHNAVI KASHYAP (vaishnavikas@umass.edu)**
SHREYA BALAKRISHNA (shreyabalakr@umass.edu)

Project #71

# Introduction and Methodology

**Problem Statement:** Medical transcription datasets often suffer from **class imbalance**, where certain categories are overrepresented while others are underrepresented.This imbalance can lead to inaccurate classifications, particularly for the underrepresented classes.

**Proposed Solution:** Use **Large Language Models (LLMs)** such as ClinicalBERT, ALBERT, T5, GPT-Neo, and GPT-J and compare with traditional methods like SMOTE for **data augmentation** to generate synthetic samples for minority classes.

**Classification Approach:** After augmentation, apply **CNN** (Convolutional Neural Networks) and **CNN + BiLSTM** (Bidirectional Long Short-Term Memory) for **classification** to evaluate the impact of augmented data on model performance.

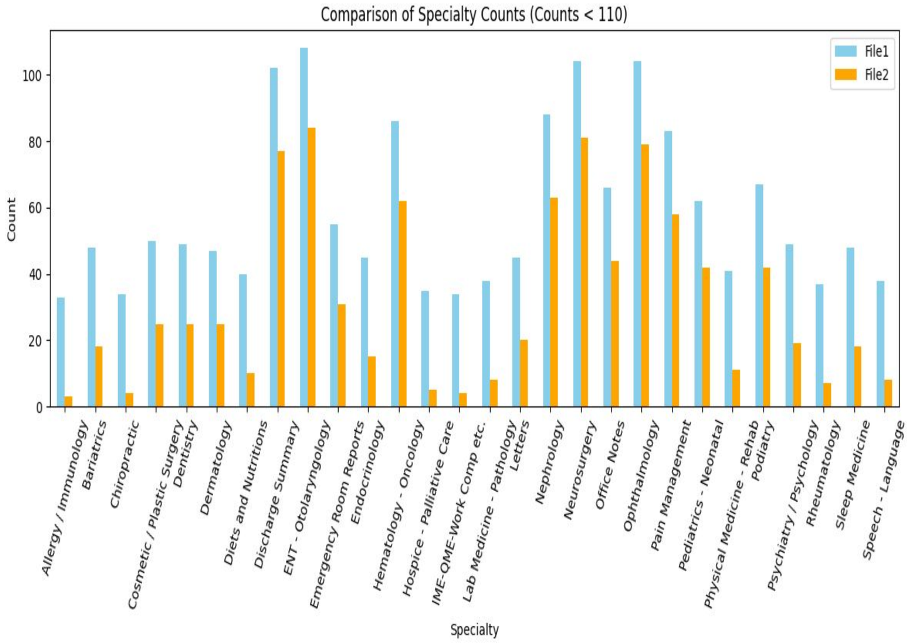
**Common Augmentation Strategy:**

**Synonym Replacement:** Up to 10 randomly selected words in each text sample were masked and replaced using model predictions.

**Strategic Masking:** Additional masks were placed at specific positions, such as the second and second-to-last words (ClinicalBERT) or at the beginning and end of the text (ALBERT), to encourage diverse replacements.

**Sampling Rules:**

- Classes with fewer than 20 samples were augmented to have 30 new samples.
- Classes with 20 to 110 samples received 25 new samples.
- Classes with more than 110 samples received only 1 additional sample




Class distribution after augmentation with ClinicalBERT

# Results and Comparison with previous work

The experimental results demonstrate the superior performance of **ALBERT** across both CNN and CNN+BiLSTM architectures, achieving the highest accuracy (**94.33%** with CNN+BiLSTM and **94%** with CNN), recall, and F1 scores, significantly outperforming traditional methods like SMOTE, which struggled with accuracy (51.94% and 53%).

Other LLMs, such as ClinicalBERT, GPT-Neo, and T-5, also showed promising results, with ClinicalBERT achieving high accuracy (91.95%) and GPT-Neo excelling in precision (92.09%), though they did not surpass ALBERT in overall performance.

**<u>On Comparing with previous work:</u>**
Given the significant class imbalance in this dataset, there has been limited work with this dataset. Previous studies on data augmentation with LLMs for this dataset achieved accuracy in the range of 65-75%. Our implementation demonstrates an impressive improvement, with an accuracy of 94%.

| | **CNN** | | | |
|---|---|---|---|---|
| | **Accuracy** | **Recall** | **Precision** | **F1-Score** |
| **ClinicalBert** | 0.93 | 0.939 | 0.944 | 0.940 |
| **ALBERT** | 0.94 | 0.949 | 0.952 | 0.948 |
| **T5** | 0.81 | 0.818 | 0.906 | 0.833 |
| **GPT-Neo** | 0.83 | 0.831 | 0.920 | 0.849 |
| **SMOTE** | 0.53 | 0.532 | 0.830 | 0.538 |

| | **CNN+BiLSTM** | | | |
|---|---|---|---|---|
| | **Accuracy** | **Recall** | **Precision** | **F1-Score** |
| **ClinicalBert** | 0.91 | 0.947 | 0.943 | 0.972 |
| **ALBERT** | 0.94 | 0.972 | 0.947 | 0.960 |
| **T5** | 0.75 | 0.823 | 0.875 | 0.848 |
| **GPT-Neo** | 0.77 | 0.794 | 0.964 | 0.870 |
| **SMOTE** | 0.51 | 0.483 | 0.967 | 0.645 |

# Conclusion

**LLM Superiority:** ALBERT and ClinicalBERT outperformed other models, achieving high accuracy (up to 0.94) and F1-scores (up to 0.960) in handling imbalanced medical datasets.

**Model Performance:** Both CNN and CNN + BiLSTM models performed well with LLM-based augmentation, delivering strong classification results.

**T5 and GPT-Neo Insights:** T5 and GPT-Neo showed moderate effectiveness, highlighting the need for task-specific fine-tuning.

**Limitations of SMOTE:** SMOTE improved recall and precision but struggled with accuracy and F1-scores compared to LLM-based augmentation.

**Impact:** LLMs demonstrated significant potential in addressing class imbalance, enhancing predictive reliability, and supporting healthcare research and patient care.

**Future Scope:**

- **Real-Time Data Integration:** Combine real-time medical data with LLM-based augmentation for adaptive models in dynamic clinical settings.
- **Continuous Improvement:** Use real-time feedback to fine-tune LLMs, ensuring timely, accurate predictions for healthcare professionals.

Project #77

# Geo localization

Hung Nguyen (huntnguyen@umass.edu),
John Steenbruggen (jsteenbrugge@umass.edu),
Long Vo (longvo@umass.edu)

# GeoLocator

GeoGuessing Like The Pros: Improving Image Classification with Segmentation

Motivation:

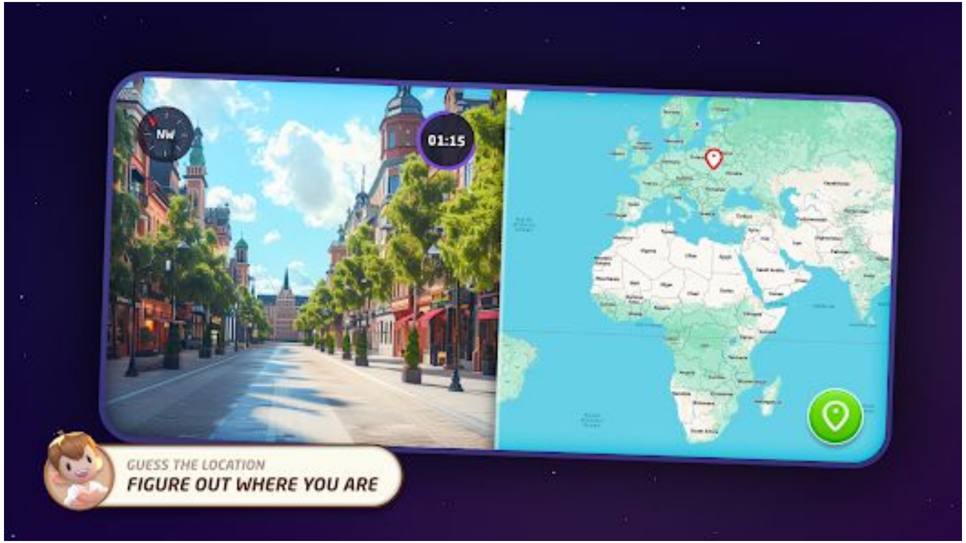
- Geo-locating images using visual features has applications in intelligence, environmental monitoring, and personal recreation (e.g., GeoGuessr game).
- Goal: Replicate the strategies of expert GeoGuessr players by using segmentation to analyze geographical features (e.g., roads, terrain, signs).

Objective:

- Enhance image classification using a Convolutional Neural Network (CNN) integrated with segmentation techniques.
- Improve geo-location prediction for images from Google Street View.

Dataset:

- 40,000 Google Street View images with a focus on rural locations, available from Kaggle.



# Results and performance

**Key Findings:**
Baseline Model:

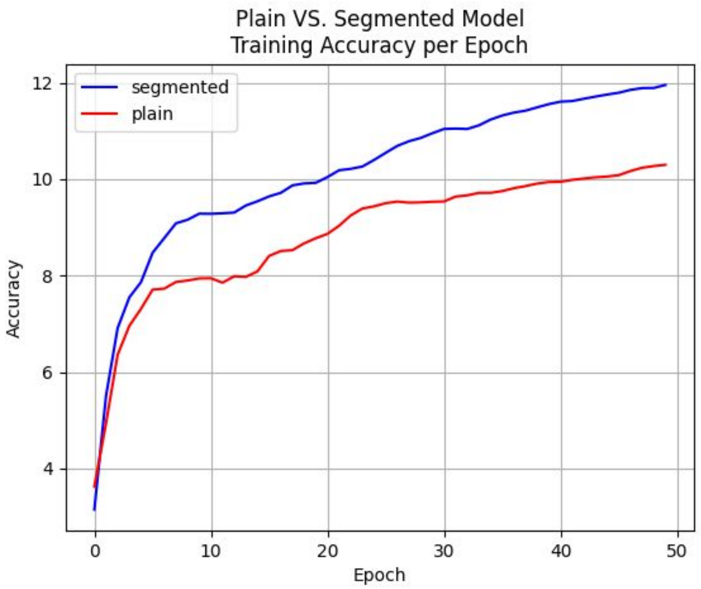
- CNN trained on whole images gives 17% accuracy in classifying images to the correct grid cell, far exceeding the random guess accuracy of 0.6%.

Segmented Model:

- Improvement expected with image segmentation, allowing better classification by focusing on distinct objects like roads, terrain, and fauna.

Comparison to Prior Work:

- Previous geo-location models (e.g., PlaNet) use global classification techniques, but segmentation allows more fine-grained, object-based analysis.
- Our approach improves geo-location prediction by leveraging segmented analysis of key objects.



# Conclusion

Challenges:

- Dataset biases towards rural environments, which lack clear infrastructure markers like signage.
- Variability in natural and man-made environments, including seasonal changes and infrastructure evolution, makes prediction harder.

Future Work:

- Further tuning of segmentation models and integration with more diverse datasets for broader geographic coverage.
- Explore other CNN architectures (e.g., ResNet, EfficientNet) and advanced segmentation techniques.

Takeaway:

- Image segmentation offers a promising path to improve geo-location tasks by capturing region-specific visual features, with potential to outperform generic models.

# Badminton shot and player movement prediction

Anmol Chokshi (achokshi@umass.edu),
Kavisha Parikh (kavishaprana@umass.edu),
**Rahasya Barkur (rbarkur@umass.edu)**

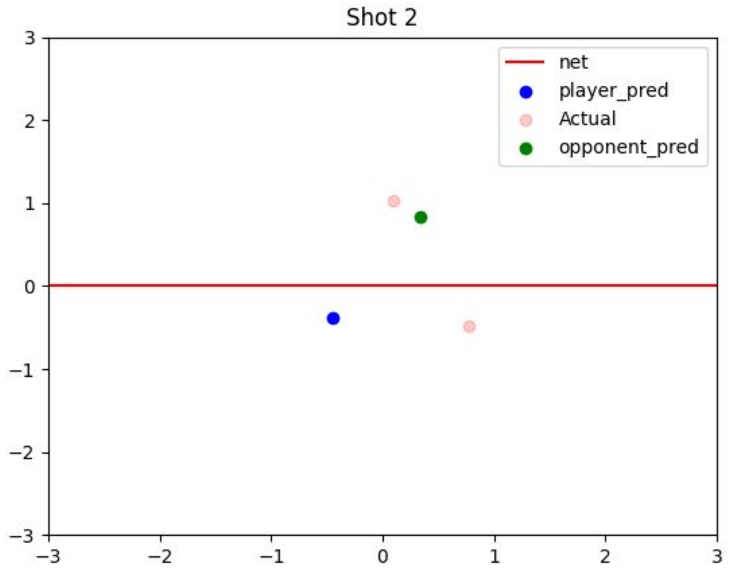
Project #80

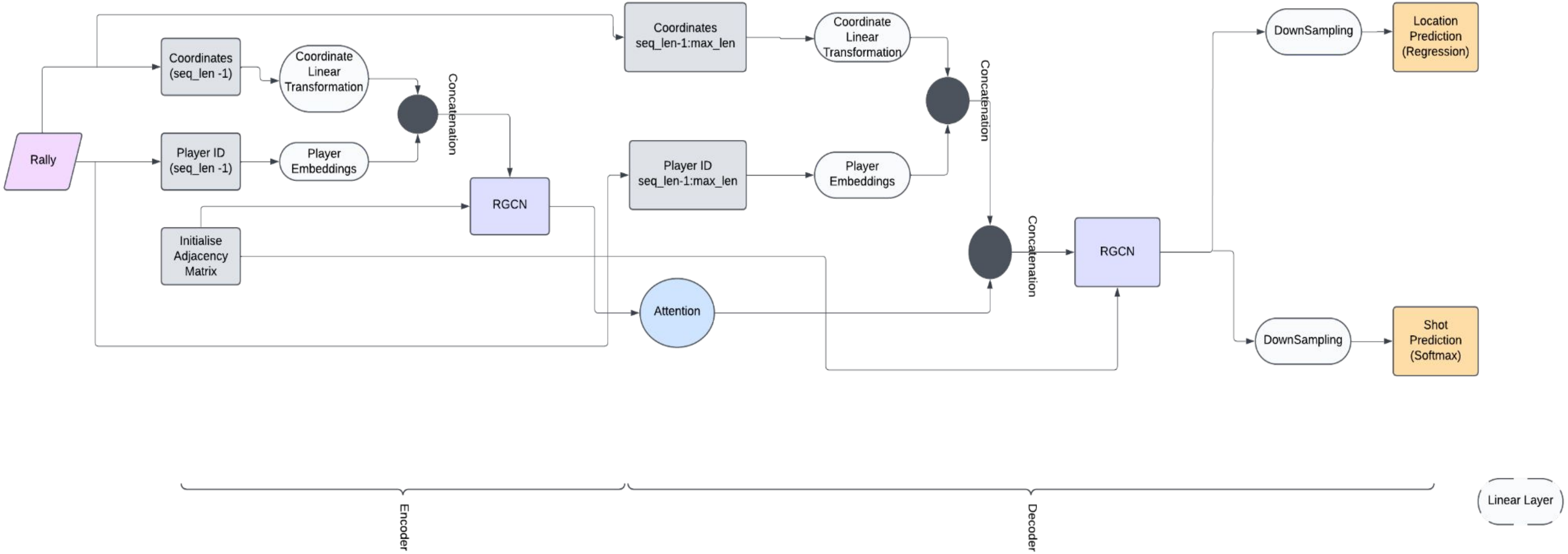
# Motivation and Summary

➔ Where will the players move on the court after each shot?
➔ What type of shot will they perform?

Approach
- Sequence to sequence models
- Graphical Models

Dataset:
Badminton Singles Dataset → 10 shot types
(x,y) coordinate of player and opponent






# RGCN -Attn (Novel Architecture)

# Results

| Model | Params | Seq Len | MSE | MAE | CE |
|---|---|---|---|---|---|
| LSTM | 10533 | 4 | 1.45 | 1.85 | 1.97 |
| | | 8 | 1.35 | 1.78 | 1.95 |
| GRU | 9445 | 4 | 1.46 | 1.83 | 2.13 |
| | | 8 | 1.37 | 1.80 | 2.34 |
| ShuttleNet | 32512 | 4 | 1.40 | 1.81 | 1.97 |
| | | 8 | 1.47 | 1.84 | 1.96 |
| Transformer | 14608 | 4 | 1.41 | 1.84 | 1.99 |
| | | 8 | 1.31 | 1.77 | 1.97 |
| GCN | 4021 | 4 | 3.83 | 3.14 | 2.02 |
| | | 8 | 3.53 | 3.03 | 2.00 |
| R-GCN | 7249 | 4 | 1.64 | 1.99 | 2.06 |
| | | 8 | 1.41 | 1.80 | 1.97 |
| DyMF | 16027 | 4 | 1.28 | 1.76 | 1.96 |
| | | 8 | 1.14 | 1.63 | 1.96 |
| RGCN-Attn | 8881 | 4 | 1.35 | 1.78 | 1.97 |
| | | 8 | 1.22 | 1.67 | 1.95 |

Table 1. Performance metrics for different models across sequences.

- Mean Square Error (MSE) and Mean Absolute Error (MAE) for Player Location prediction
- Cross Entropy for Shot Type Predictions

Conclusion:

- ❖ Designed RGCN- Attn with almost half the parameters as compared to state of the art DyMF with competitive performance.
- ❖ Computationally inexpensive model with comparable performance.

# Immersive audio

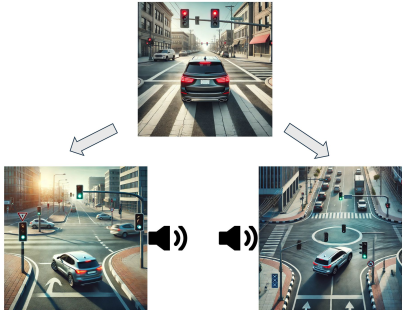
Lixing Fang (lixingfang@umass.edu),
Qinhong Zhou (qinhongzhou@umass.edu),
Sunli Chen (sunlichen@umass.edu)

Project #83

# Motivation and snapshot

**Motivation**: Existing datasets often lack multi-channel audio or fail to maintain consistent spatial and temporal correspondence between audio and video components, limiting their utility for training models designed for realistic and synchronized audio-video generation.

**Approach**: We leverages state-of-the-art models, including video-language models, open-vocabulary object detection, segmentation, and consistent depth estimation, to identify, track, and synthesize audio for sound sources within videos.

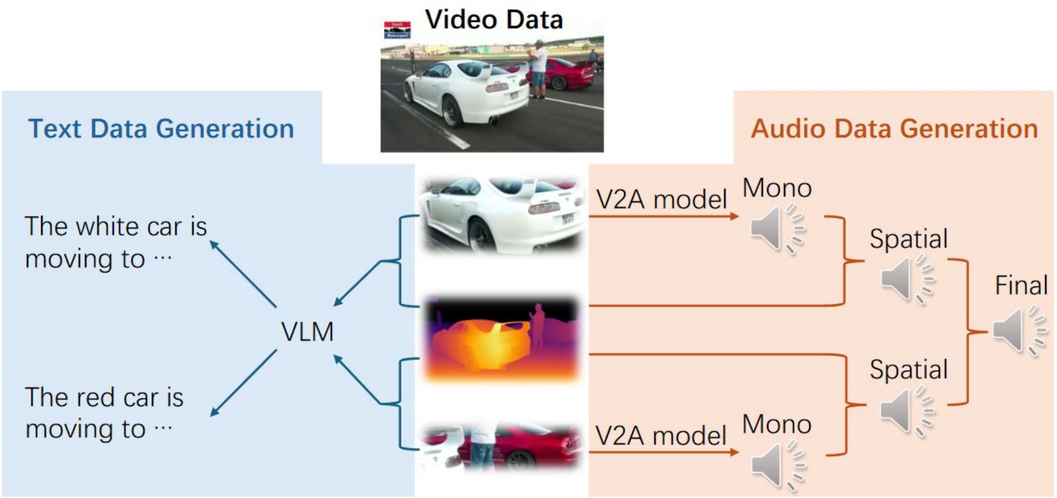


**Goal**: By combining pixel-level segmentation, depth information, and motion tracking, the pipeline generates multi-channel spatial audio that accurately corresponds to video content in both time and space

# Data & Model

**Data Generation:**

- Motion from video
  - Object detection + tracking
  - Metric depth estimation (assumed camera intrinsic)
- Prompting with VLM
  - Which objects are making sounds?
  - How is each object moving?
- Sound Composition
  - Video to Audio with pretrained LDM models
  - Impulse Response based on distance & direction
  -



# Core contributions

This work addresses the challenges of synthesizing high-quality spatial audio-video data enriched with textual context, including:

- **Data Generation Pipeline**: a comprehensive pipeline for creating spatial video data with synchronized audio and text, unlike prior approaches that often result in low-quality spatial outputs.
- **Large-Scale Dataset**: an extensive spatial audio-video dataset paired with textual descriptions, serving as a critical resource for advancing multimodal research and training models in this domain.

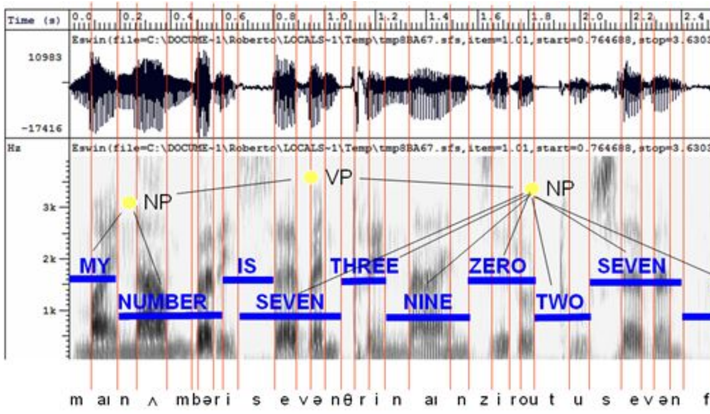
| Dataset | Spatial Sound | Text Description | Spatial Annotations |
|---|---|---|---|
| VGGSound [1] | No | Yes | No |
| STARSS [11] | Yes | Yes | No |
| Youtube-360 [8] | Yes | No | No |
| Ours | Yes | Rich | Yes |

# FusionASR: Conformer and Language Model Integration for Automatic Speech Recognition

**Jiageng Liu (jiagengliu@umass.edu),**
Zhehuan Chen (zhehuanchen@umass.edu),
Wenjun Liu (wenjunliu@umass.edu)

**Project #85**

# Introduction




**Overview of ASR:**

- **Automatic Speech Recognition (ASR)** converts speech into text.
- Applications: Virtual assistants, transcription, accessibility tools.
- Challenges: Variability in speech patterns, accents, rates, and noise.

**Traditional vs. Modern Approaches:**

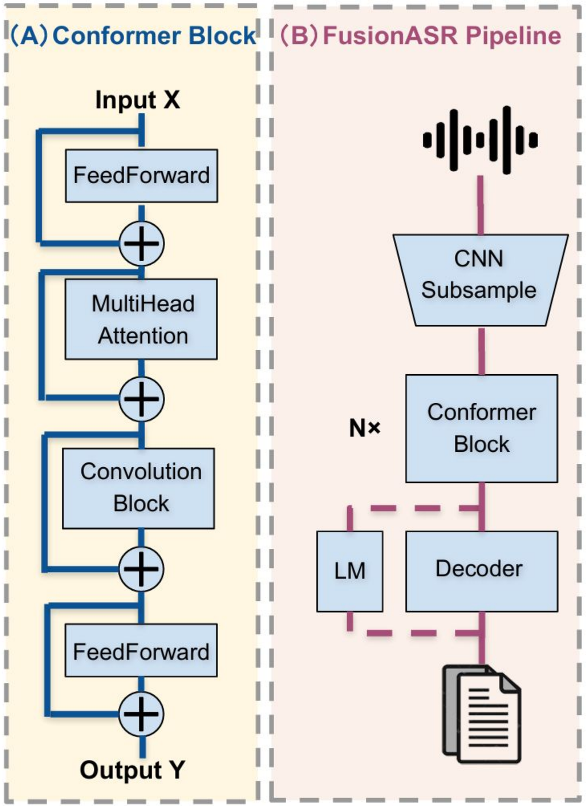
- Traditional: HMMs and GMMs struggled with complex speech dynamics.
- Modern: Deep learning (**CNNs, RNNs, LSTMs, Transformers**,) models richer features, improving accuracy.
- **Conformer**, effectively capture local/global speech dependencies using self-attention and convolution.

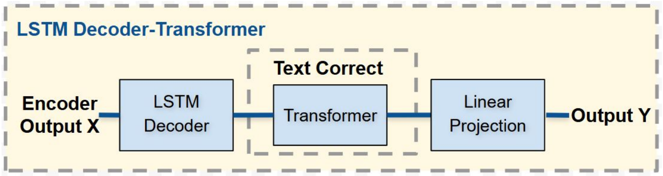
**FusionASR Innovation:**

- Combines Conformer-based encoder with varied decoders for improved transcription.
- Trained on **LibriSpeech** Datasets.
- Leverages **GPT-2 tokenizer**, **data augmentation**, and **language modeling** for semantic richness boosts.

# Method

- **Conformer Encoder:**
  - **Feed-forward,**
  - **Multi-head self-attention – global dependency**
  - **convolutional modules – local feature**
  - **residual connections**
- **Decoder:**
  - **LSTM Decoder**: Processes input sequences to predict outputs using memory-based gating mechanisms.
  - **Linear Decoder:** Maps encoded features to outputs via simple linear transformations.
  - **Transformer Decoder:** Uses self-attention for parallel, context-aware sequence processing.
  - **LSTM Decoder-Transformer:** Combines LSTM with Transformer for semantic sentence correction and error reduction.
- **Improvements:**
  - **Data Augmentations:** Enhances model robustness using SpecAugment techniques like masking and time warping.
  - **Tokenized ASR Model:** Reduces errors by using GPT-2 token-level decoding with Byte Pair Encoding.
  - **Hybrid ASR System:** GPT2 as scorer.



# Experiment Result and Conclusion

- **Experiment Result:** Compare to current SOTA method. (Table 1)
- **Ablation Study:** Each Module (Table 2); Hyperparameters (Table 3~5)
- **Conclusion:**
    - Explore **Different Decoders**, **Tokenization**, **Hybrid Architecture**.
    - Reach 12.6% WER, exceeding current **SOTA** method.
    - Further work needed in **LM Fusion**, **Larger Scale**, **Datasets**.

1

| Methods | test-clean | test-other |
|---|---|---|
| QuartzNet [12] | 15.1 | 26.7 |
| ContextNet [8] | 13.7 | 20.2 |
| Conformer [7] | 12.9 | 18.8 |
| FusionASR(Ours) | **12.6** | **17.6** |

2

| Methods | test-clean | test-other |
|---|---|---|
| w/o GPT-2 Transformer | 12.9 | 18.8 |
| w/o Tokenization | 45.4 | 60.3 |
| Linear Decoder | 13.1 | 18.4 |
| Transformer Decoder | **12.5** | 18.0 |
| w/o Data Augmentation | 20.7 | 29.2 |
| FusionASR(Ours) | 12.6 | **17.6** |

3

| Head Number | test-clean | test-other |
|---|---|---|
| 2 | 18.9 | 20.8 |
| 4 | **12.6** | **17.6** |
| 8 | 12.8 | 18.0 |
| 16 | 12.8 | 17.9 |

4

| Number of layers | test-clean | test-other |
|---|---|---|
| 8 | 20.1 | 32.5 |
| 12 | 13.9 | 23.0 |
| 16 | **12.6** | **17.6** |

5

| Window size | test-clean | test-other |
|---|---|---|
| 1/4 | 78.2 | 81.7 |
| 1/2 | 25.2 | 40.1 |
| 1 (whole sequence) | **12.6** | **17.6** |

# Attribute driven person re-id for passenger counting using spatio-temporal patterns in data scarce scenarios

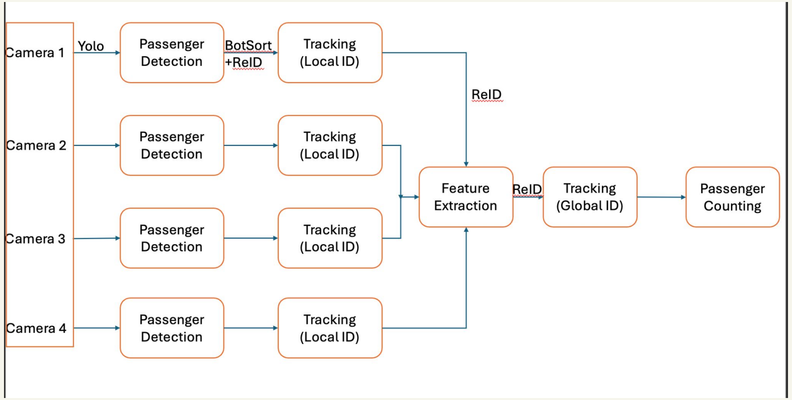
Nthenya Kyatha (mkyatha@umass.edu),
Sanuratu Koroma (skoroma@umass.edu)

Project #88

# Project Overview

## Motivation/Background

Most people travel by bus, but knowing the number of people that use the buses, areas in which they live, the time the majority of people commute is a challenge for developing countries. This makes it hard to properly allocate resources to areas where they are needed.




## Solution

Luckily, the newly rolled out E-buses come with CCTV cameras. Our proposed solution is to use state-of-the-art computer vision models, specifically YOLO, BoT-SORT and OSNet to count the number of passengers boarding and alighting.

University of Massachusetts Amherst

# Results





## Challenges

- Passengers going in very close together resulted in tracking only the one at the back.
- Overcrowding in the buses made it difficult to accurately define the boundary.
- Line-based counting or region-of-interest (ROI) counting introduced double counting
- Due to similar clothes, poses and occlusions cause ID mis-assignment.



University of Massachusetts Amherst

# Conclusions

## Contributions

- Examined the performance of YOLO on out-of-distribution data, focusing on darker-skinned people.
- Examined the performance of  Re-ID models in high-density environments with occlusions and poor lighting

## Future-work

- Enhancing model robustness through additional fine-tuning on diverse, domain-specific datasets and integrating advanced tracking algorithms to mitigate counting inaccuracies in dynamic environments.
- Explore other counting techniques that don't include ROI



University of Massachusetts Amherst