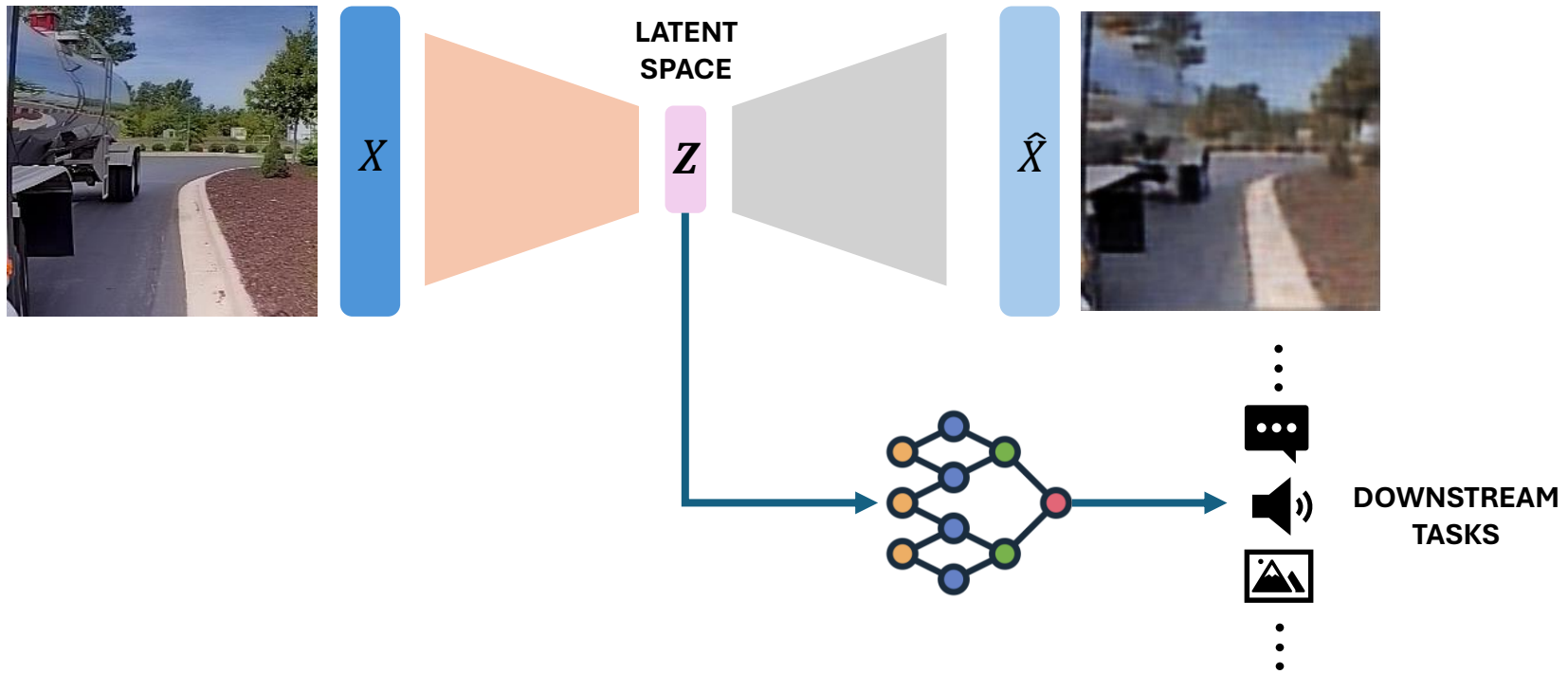
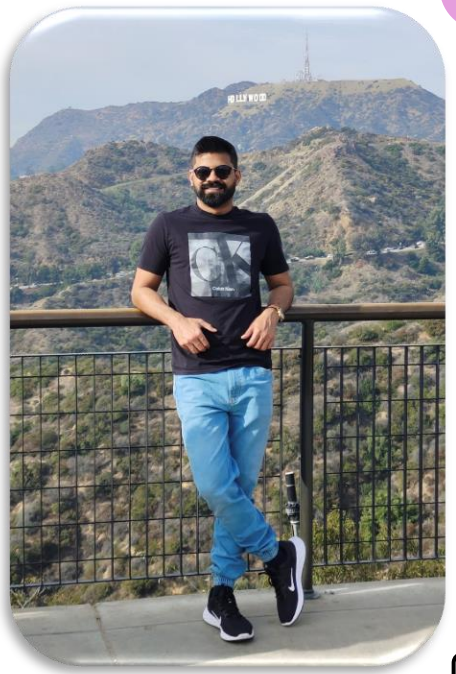


Generative Model based Computer Vision Application for Driver Assistance

- Meeshawn Marathe

April 15, 2024

About Me



Control Systems

Embedded Systems (Arduino, Rasp Pi, TI C2000)

Image Signal Processing

Motor Control

Software Engg.

Deep Learning

Audio Signal Processing

Generative AI

Recommender Systems

Multimodal AI

Classical ML

Deep Learning

NLP

Computer Vision

Knowledge Distillation

Information Retrieval

Reinforcement Learning

Speech Recognition

2017

2017-2021

MathWorks

2021-2022

09/2022 - Present

DEARBORN COLLEGE OF ENGINEERING & COMPUTER SCIENCE

May – Aug 2023

10/2023 - Present

Stoneridge

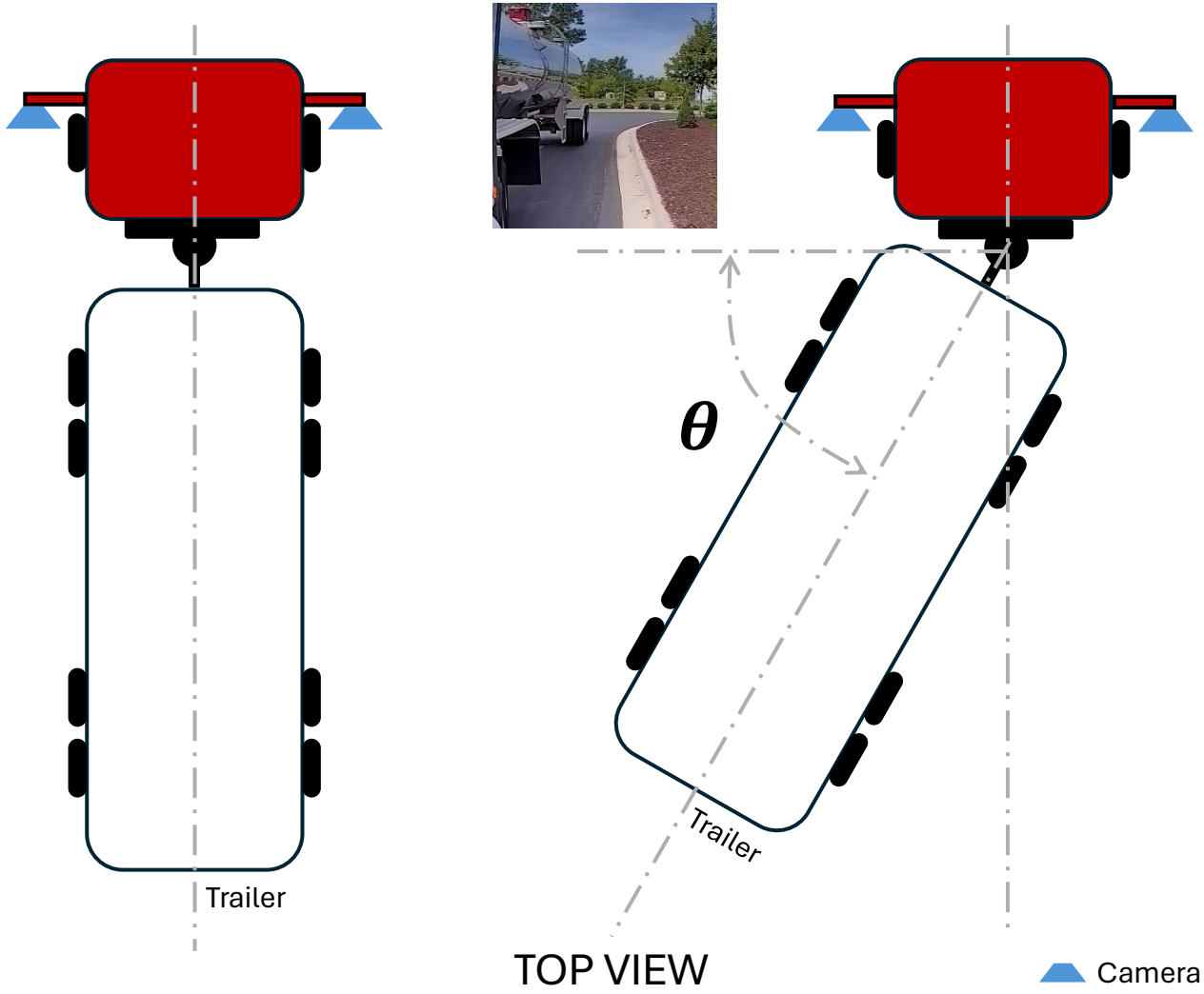
SITUATION: Topic Background - Current Internship Work

MirrorEye® Camera Monitor System

Stoneridge

TASK: Topic Background - Problem Statement

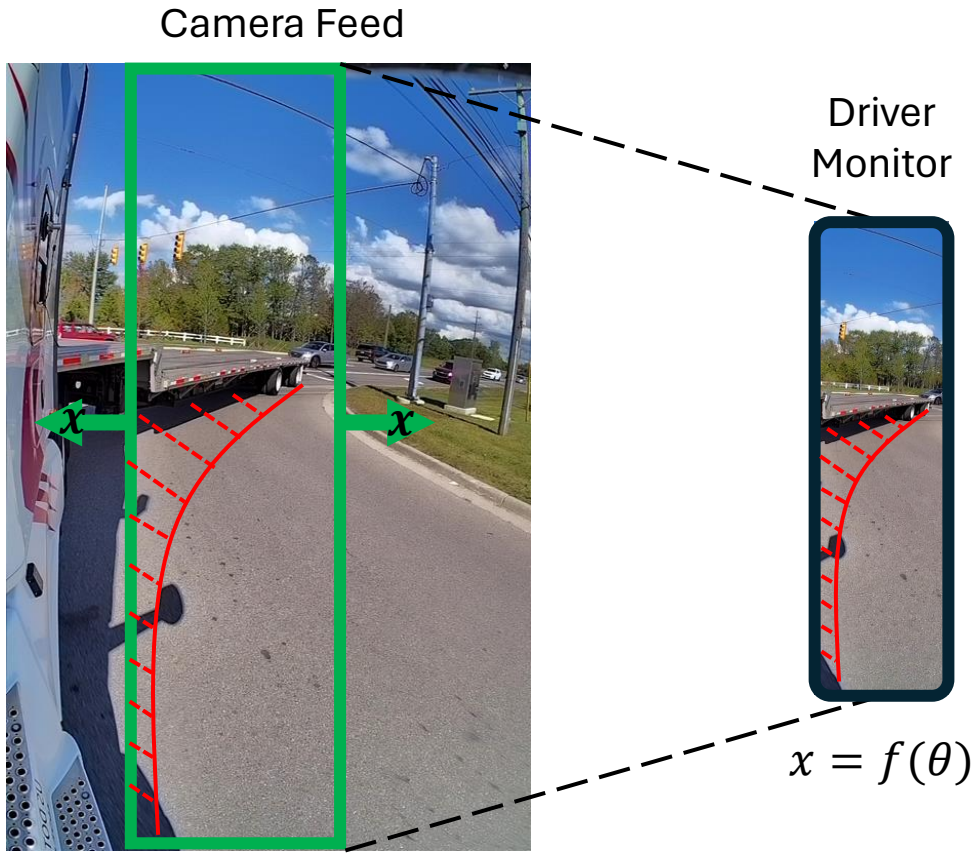
- Estimate Trailer Angle $\boldsymbol{\theta}$ under different conditions using video feed from MirrorEye CMS
- Build an Estimation Algorithm robust to these conditions:
  - Lighting Conditions:
    - Dusk/Day/Night
  - Trailer Conditions:
    - Body Type
    - Body Color



SITUATION: Topic Background - Why this problem?

- Why Trailer Angle Estimation?
  - Sub-task for other algorithms:
    - Trailer Auto-Panning
    - Trailer Safety Path

- Contribution - **Personal Initiative**: Why select this problem?
  - Generative approach can potentially replace 3 existing models:
    - Wheelbase Detection: Fails at night
    - Trailer End Detection
    - Tape Detection
  - Robustness against Lighting conditions, trailer types
  - Reduce latency on edge device
  - Reduce Data Collection Effort



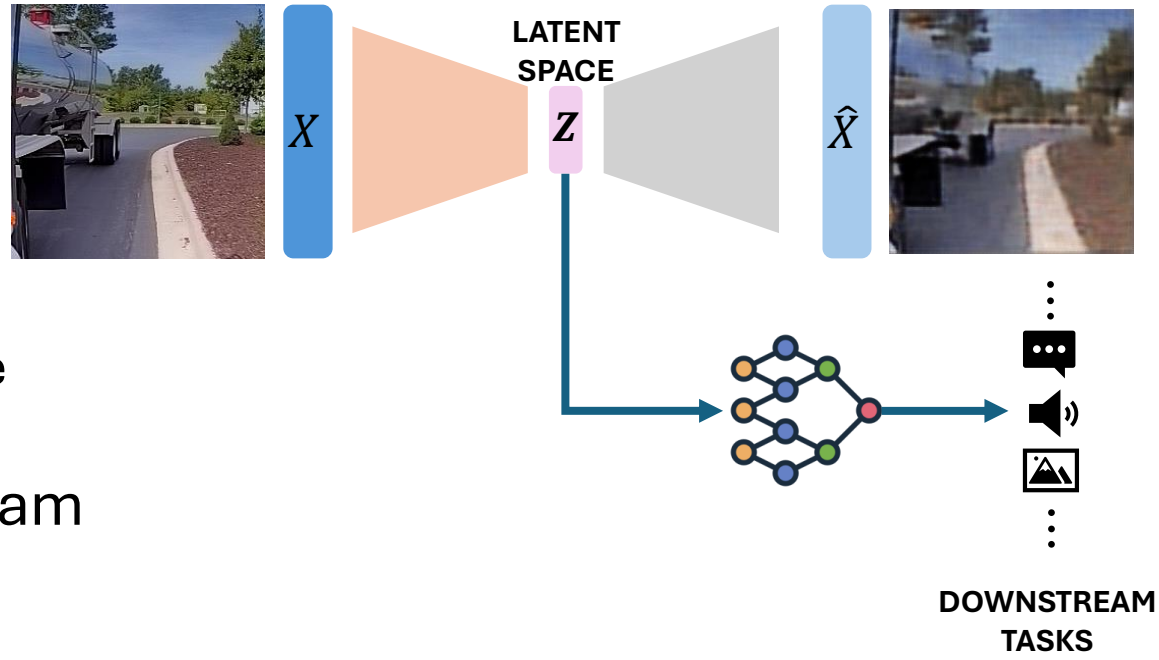
Auto-Panning: Adjustable Field of View
Trailer Safety Path: Trajectory of Trailer-end while turning

ACTION: Identified Generative Methods for the problem

Motivation

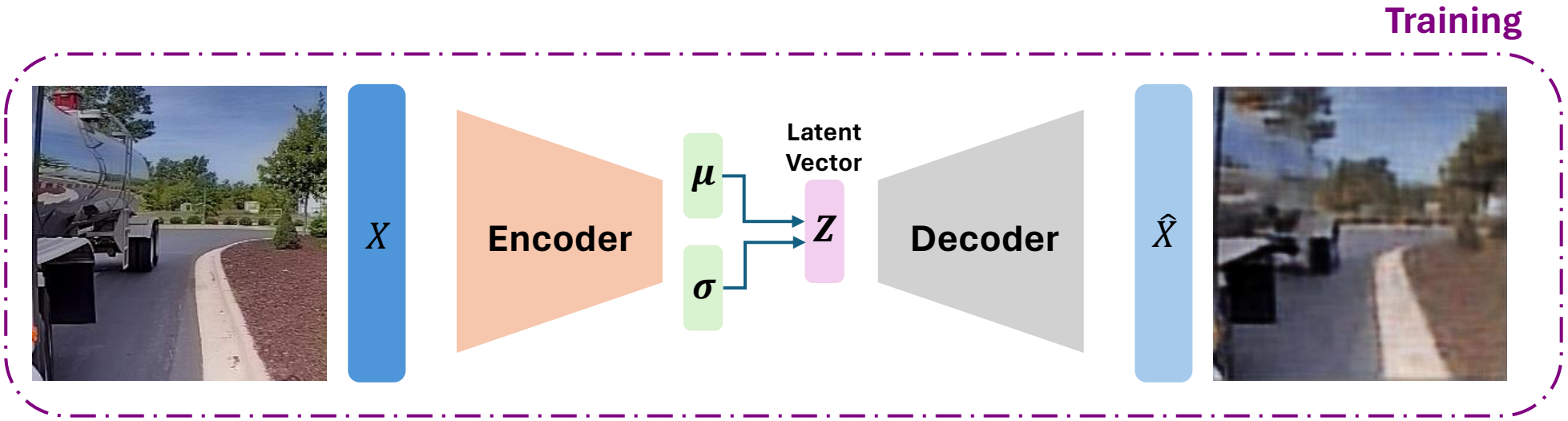
- Unsupervised Learning
  - No labels → Cheaper
  - Dimensionality Reduction
  - Underlying hidden structure → Inference

- Latent Space Arithmetic → Downstream Tasks:
  - Joint Embedding Prediction :
    - Information Retrieval (Ex - CLIP)
    - Text/Music/Image/Video Generation/ VLMs (Ex – DALL-E, Sora, MusicGen)
  - Classification/Object Detection/Segmentation



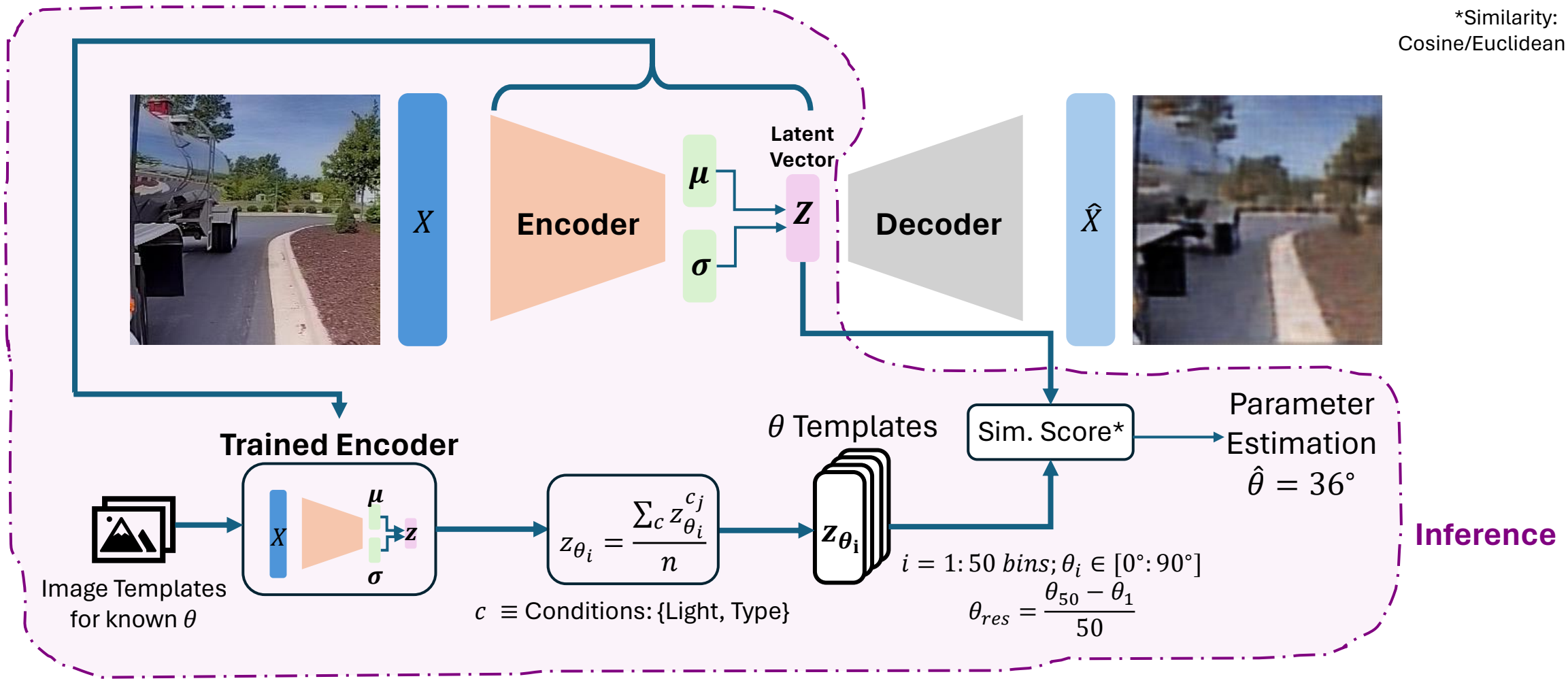
Takeaway: *"Generative methods help uncover the underlying DNA from images helpful for tasks of interest"*

ACTION: Workflow - Investigated Latent Space Arithmetic Techniques on the Trained Generative model



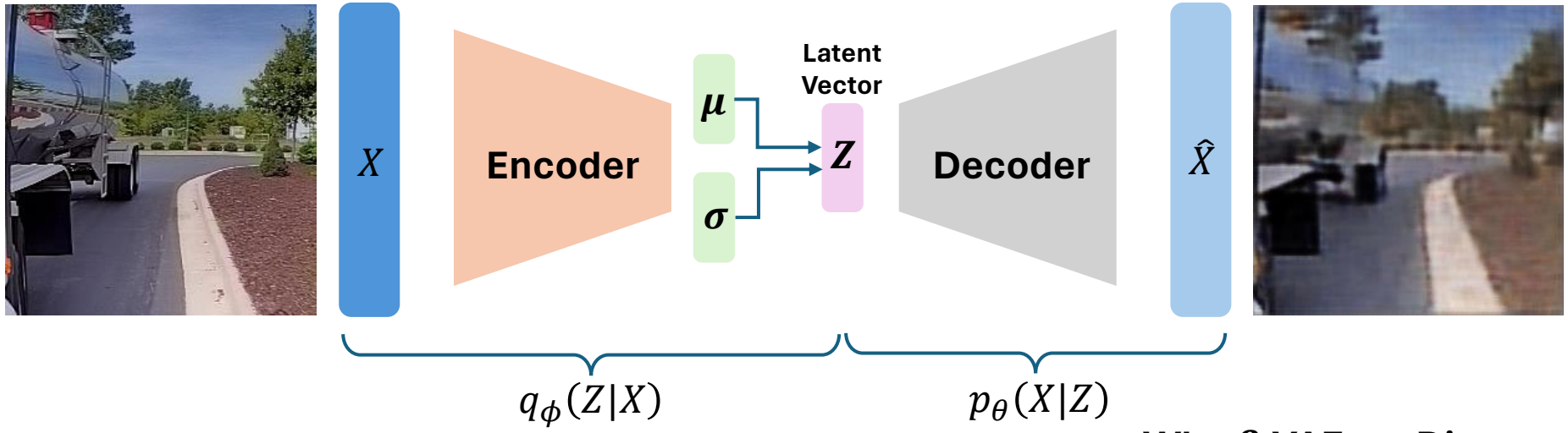
Takeaway: *"Learn a compressed latent representation of the images to help estimate the trailer angle"*

ACTION: Workflow - Investigated Latent Space Arithmetic Techniques on the Trained Generative model



Takeaway: *"Use the latent vector of a test image and compare against saved templates to estimate θ"*

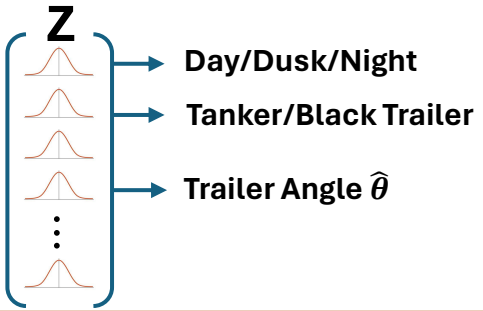
ACTION: Architecture - Identified β-VAE for Latent Representation Learning



Why VAEs?

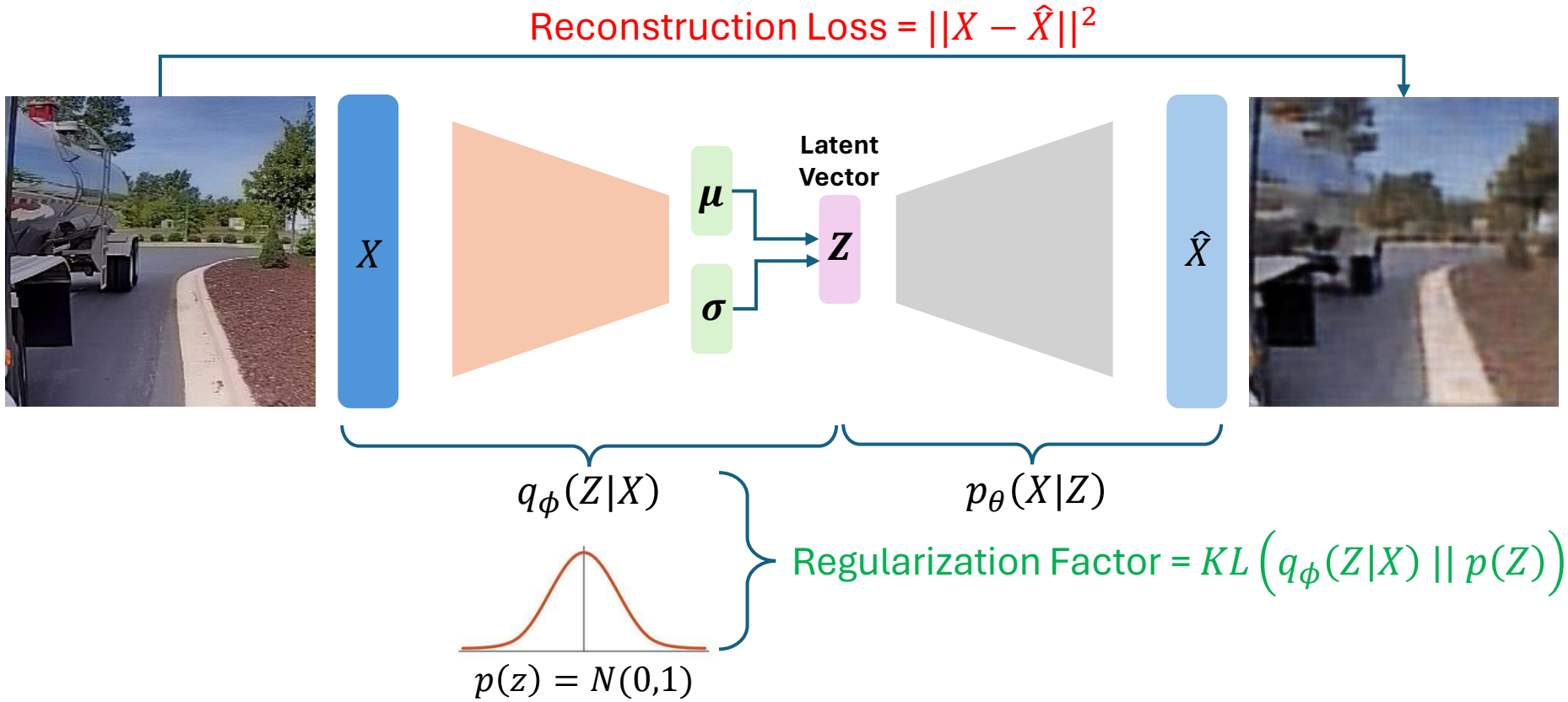
- Stochastic 'Z' instead of Fixed 'Z'
- Normal Prior: $p(z) = N(\mu = 0, \sigma^2 = 1)$
  - Avoids overfitting, promotes continuity and regularization.
  - Truly Generative ! → Generates new data

Why β-VAEs → Disentanglement



Takeaway: *"VAEs help learn compressed, but regularized latent structure helpful for the generative task"*

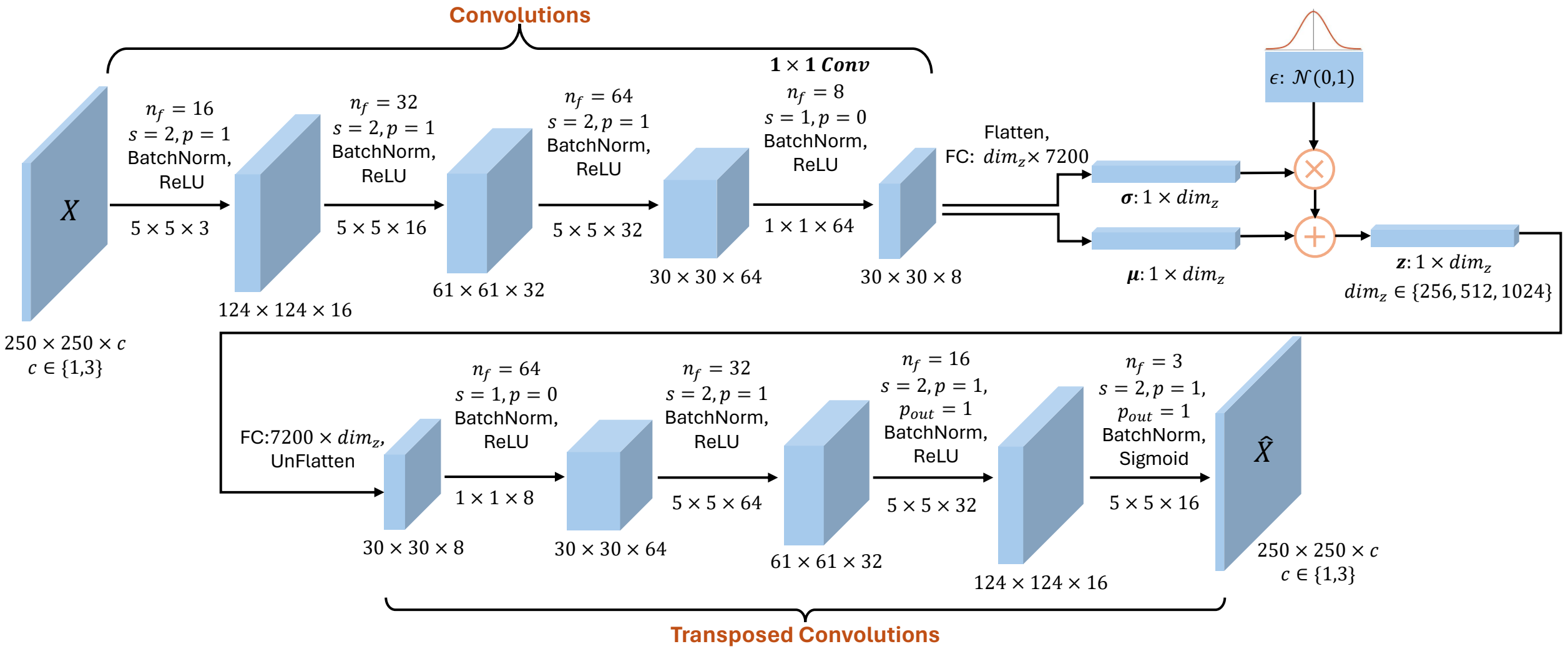
ACTION: Architecture - Identified β-VAE for Latent Representation Learning



Training Loss = Reconstruction Loss + β ·Regularization Factor

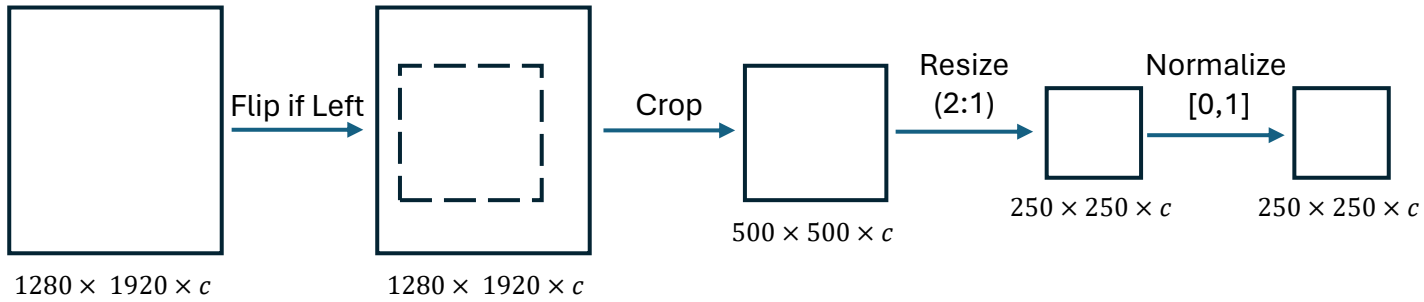
Takeaway: *"VAEs help learn compressed, but regularized latent structure helpful for the generative task"*

ACTION: Architecture: Investigated and designed the underlying neural network architecture



ACTION: Model Training – Implemented and trained the architecture in PyTorch, performed hyperparameter tuning

- Preprocessing:



- Dataset:
  - Trailer motion captured from Xylon Logs.
  - Trailer Types ∈ {white-box, black-box, oil-tanker, flatbed}. Light conditions ∈ {Day, Dusk, Night(IR)}
  - # Training Images ~ 3000 across different scenarios (balanced dataset)
  - 50 Templates averaged across scenarios, where each template angle $\theta_i \in [0°: \theta_{res}: 90°]$, where $\theta_{res} = \frac{\pi/2}{50}$
  - Torch Dataset Generator with random shuffling and transform function (preprocessing)

- Hyperparameters:
  - Batch Size$\in \{64,128\}$
  - Latent Dimension $dim_z \in \{256,512,1024\}$
  - Optimizer: Adam, Learning Rate $\alpha = 3 \times 10^{-4}$
  - Epochs $= \{50,100\}$
  - $\beta = 50$
  - Color-channels: {3: RGB, 1:Grayscale}

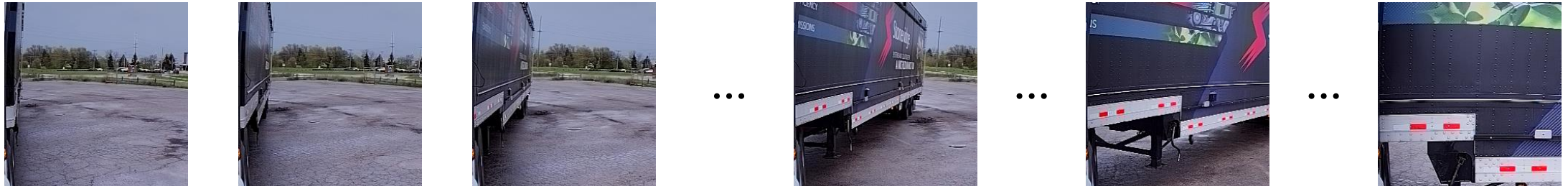
- Device: NVIDIA RTX A1000 6 GB GPU

PyTorch

ACTION: Model Training – Implemented and trained the architecture in PyTorch, performed hyperparameter tuning

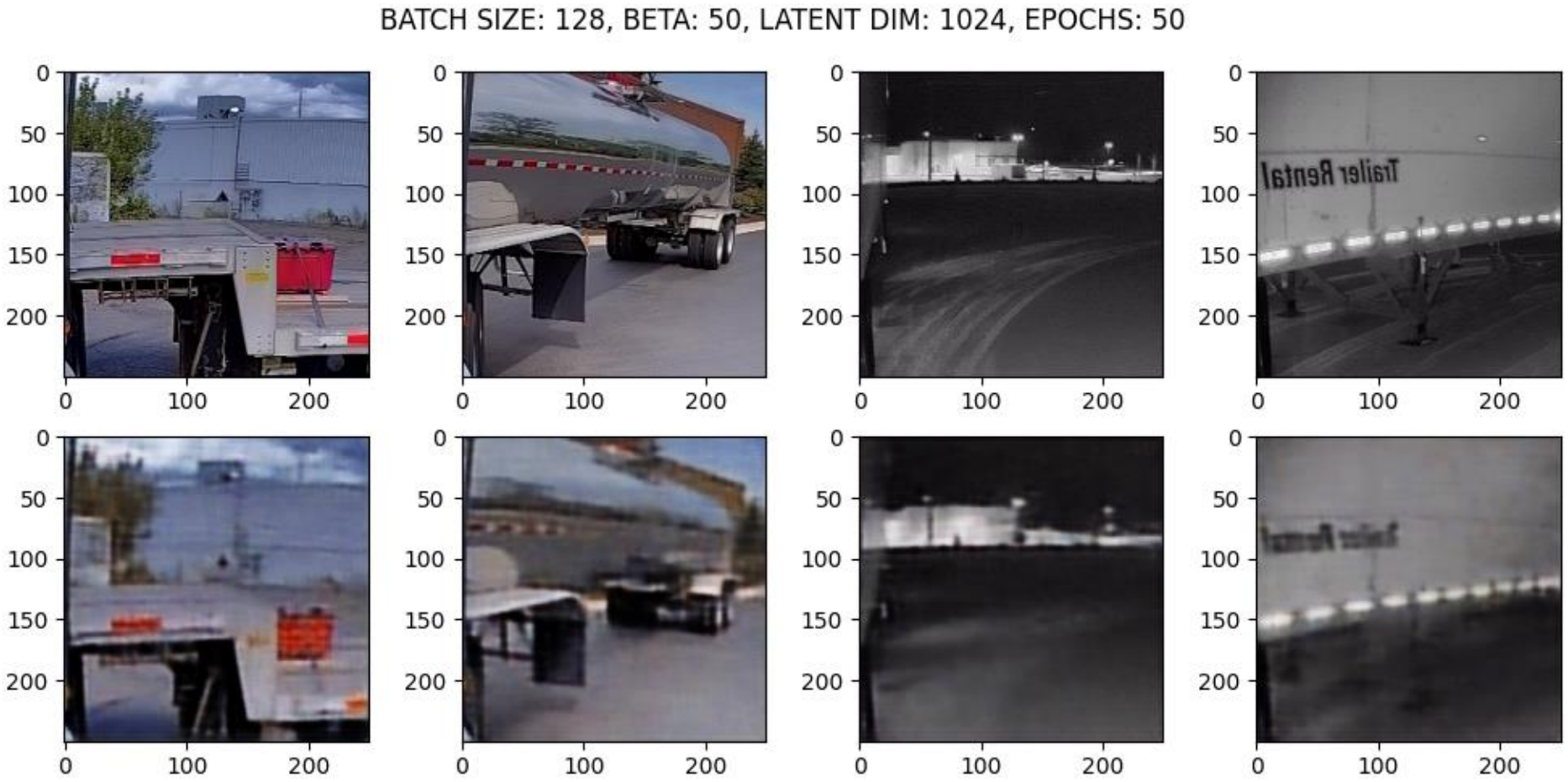
Example Templates

$\theta_1 = 0°$ θ_2 θ_1 θ_1 θ_1 $\theta_{50} = \pi/2$



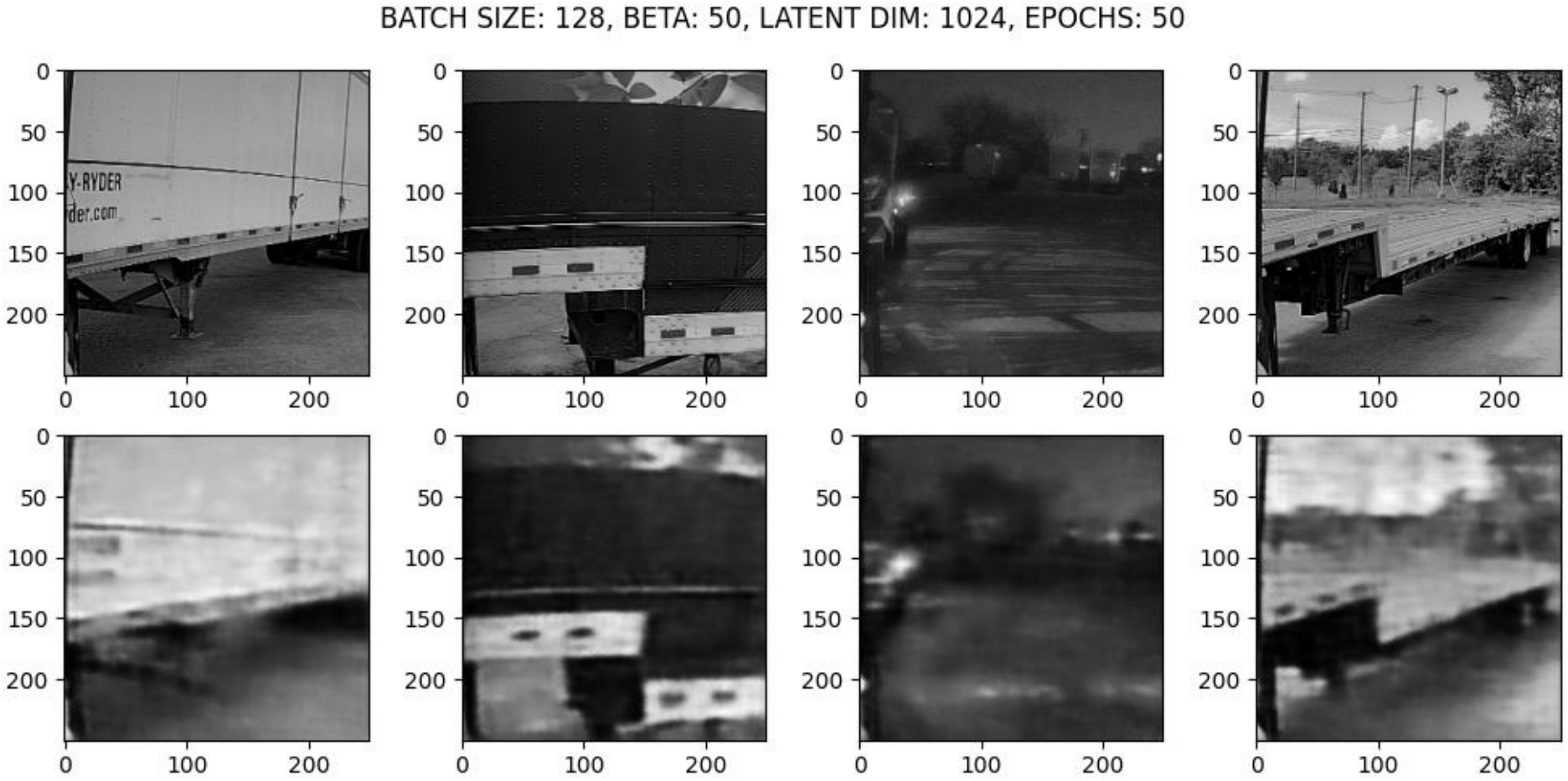
PyTorch

RESULT: Image Reconstruction using Trained VAE



RGB

RESULT: Image Reconstruction using Trained VAE



Grayscale

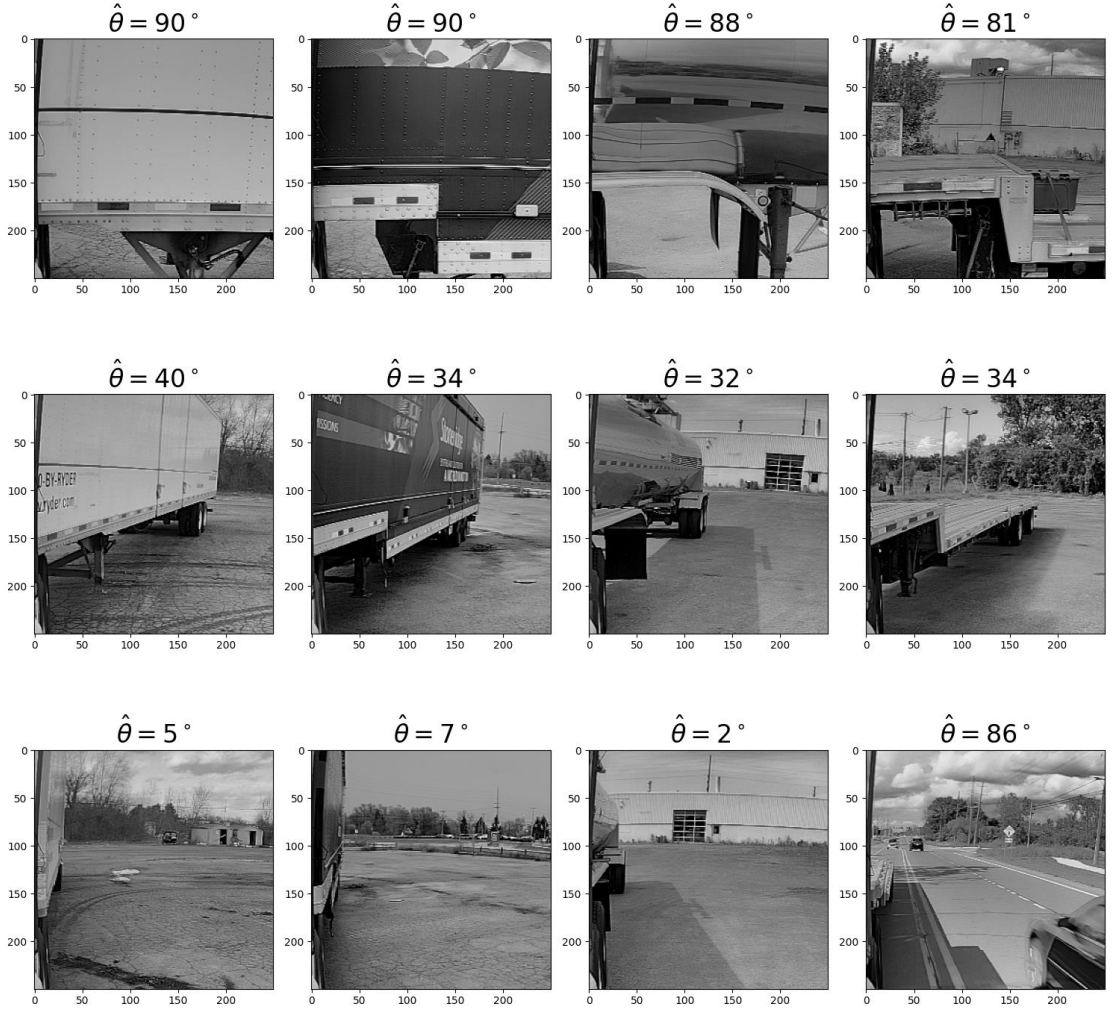
RESULT: Parameter Estimation $\hat{\theta}$ using Latent Space Arithmetic

$$Batch\ Size\ =\ 128, \beta = 50, dim_z = 1024, Epochs = 50$$

Lighting Condition: Day

Training: Grayscale Images

Increasing θ



Trailer Type

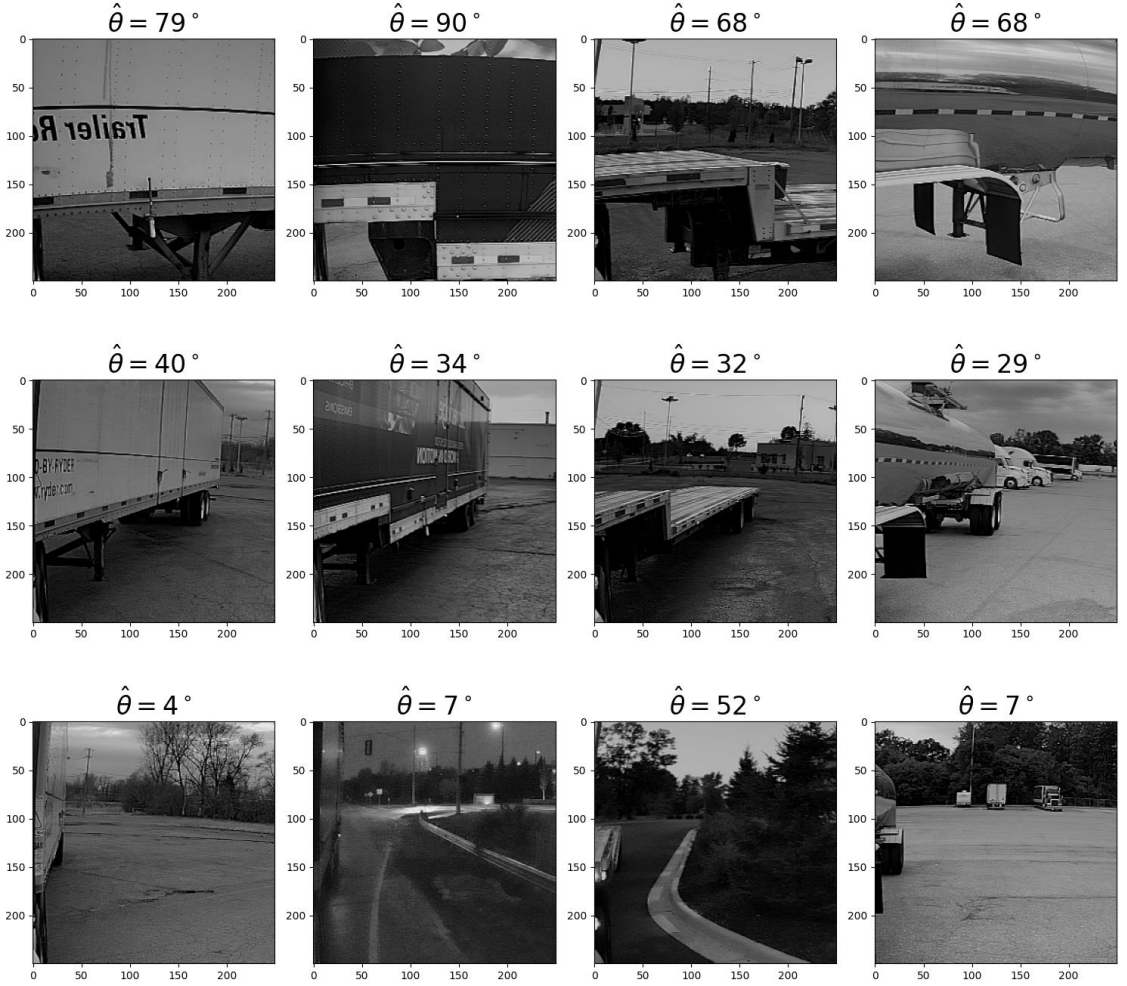
RESULT: Parameter Estimation $\hat{\theta}$ using Latent Space Arithmetic

$$Batch\ Size\ =\ 128, \beta = 50, dim_z = 1024, Epochs = 50$$

Lighting Condition: Dusk

Training: Grayscale Images

Increasing θ



Trailer Type

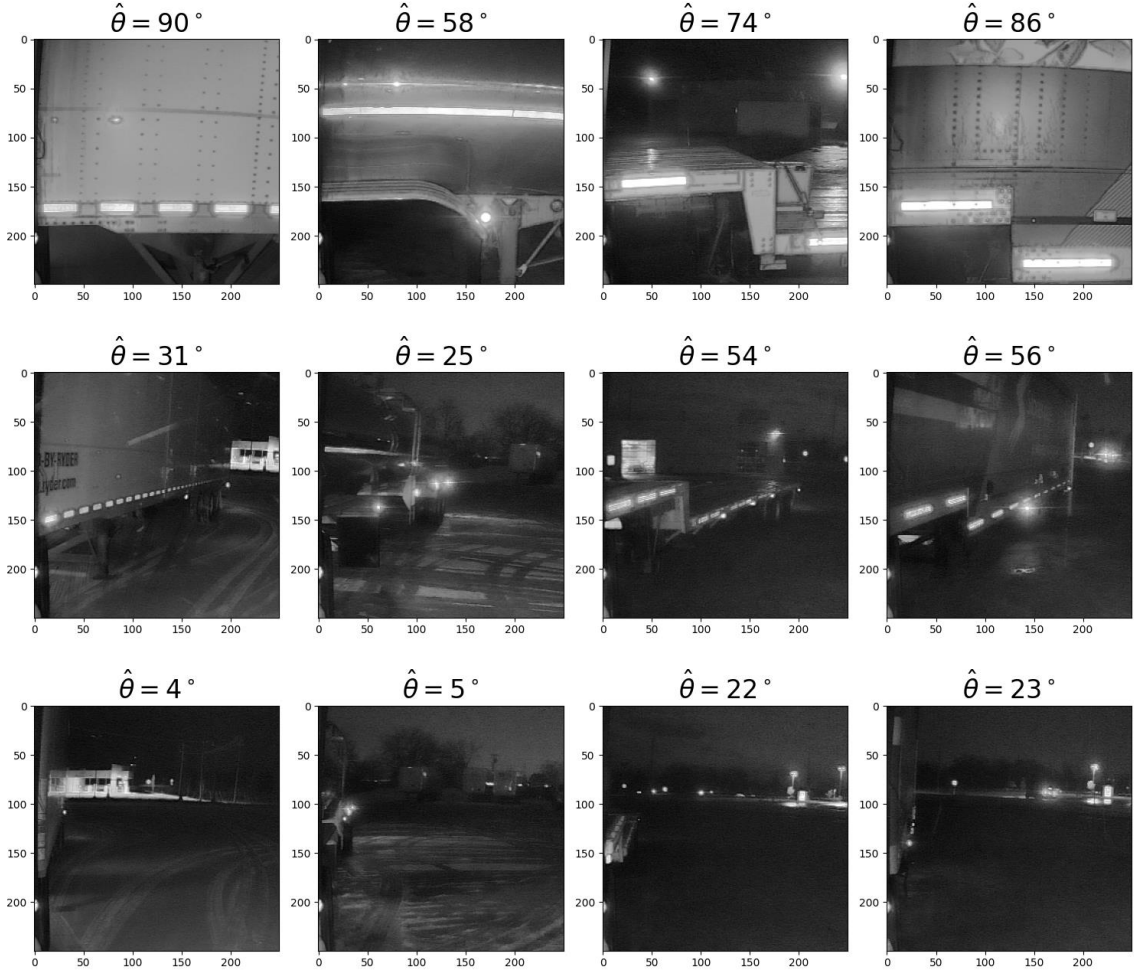
RESULT: Parameter Estimation $\hat{\theta}$ using Latent Space Arithmetic

$$Batch\ Size\ =\ 128, \beta = 50, dim_z = 1024, Epochs = 50$$

Lighting Condition: Night

Training: Grayscale Images

Increasing θ



Trailer Type

RESULT: Discussion on Trailer Angle Estimation Performance and Model Training

- **Dimensionality Reduction:** Compression achieved through VAE- RGB $\approx 1:200$, Grayscale images $\approx 1:60$
- **RGB Vs Grayscale Training**: Grayscale better for night conditions
- **Latent Dimension Selection** ($dim_z = 1024$ vs $dim_z = 512$):
  - Higher latent-dim works better for night conditions. Comparable Day/Dusk performance
  - Lower-bound on loss decreases with increase in latent dim size
- **Batch-Size Selection** (128 vs 64): Higher batch size works better for night conditions
- **Training Epochs:** Training loss tends to saturate beyond 50 epochs. Trained up to 150 epoch to check for double-descent.
- NOTE:
  - Templates averaged across all trailer types, except for *Flatbed*. This accounts for the inferior performance on Flatbed. Averaging across more scenarios might improve the performance.
  - The length of all the trailers were assumed to be the same. This accounts for slight performance issues across trailer types.
  - No Dropout layer added since regularization already provided by KL loss term. Slightly worse with Dropout.
  - Robust to reflective surfaces , such as oil tankers.
  - Scope for improving night-time performance. Otherwise, the robust across different scenarios.

RESULT: Highlights/Challenges/Lesson Learned

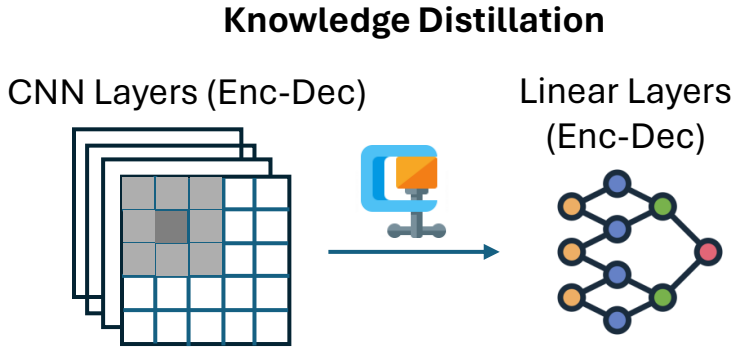
- **Highly Collaborative Effort:** Dealing with multiple stakeholders including a team of Systems Engineers for Data Collection.
- **Problem Solving:**
  - **Personal Initiative:** Unique approach to apply generative methods for solving a computer vision problem targeting automotive driver assistance applications.
  - **Direction:** Yan LeCunn stressed the importance of working in latent space in his latest lecture at NYU on 03/24/2024 on "*Do large language models need sensory grounding for meaning and understanding*". I am in the right direction!
  - **Model Selection & Training:** Issues with CUDA out of memory, hyperparameter tuning and neural architectural search
- **Impact:**
  - **Data Annotation Efforts:** No cumbersome labelling/annotation required
  - **Robustness:** Fairly robust in different conditions.
  - **Potential for Synthetic Data Generation**: Reduced Data Collection effort– very useful
  - **Latency:** Reduced latency on edge device due to latent space arithmetic technique
  - **Great Alternative:** Potential to replace 3 existing models: Wheelbase Detection, Trailer End Detection, Tape Detection
  - **Nice proof of concept:** Serves as a good prior for the team to extend the work to new height.

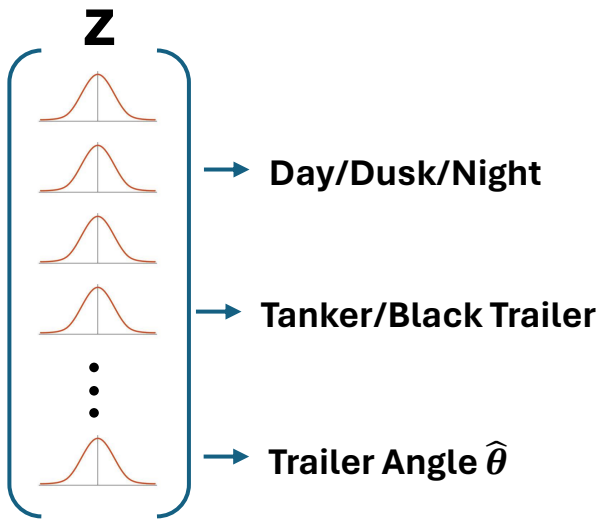
Future Work

- Model Compression:
  - Will further reduce inference latency on the edge device:
    - Knowledge Distillation
    - DepthWise & PointWise Convolution Layers using 1x1 Conv.
    - Quantization: Fixed Point Inference

- Synthetic Dataset Generation
  - Latent Space Disentanglement of → Generate Trailer images with different:
    - Trailer Angle
    - Lighting Condition
    - Trailer Body, Skin
  - Saves Effort/Resources (Data Collection)

- Architectural Search:
  - Convolutional Block Layers
  - Skip Connections

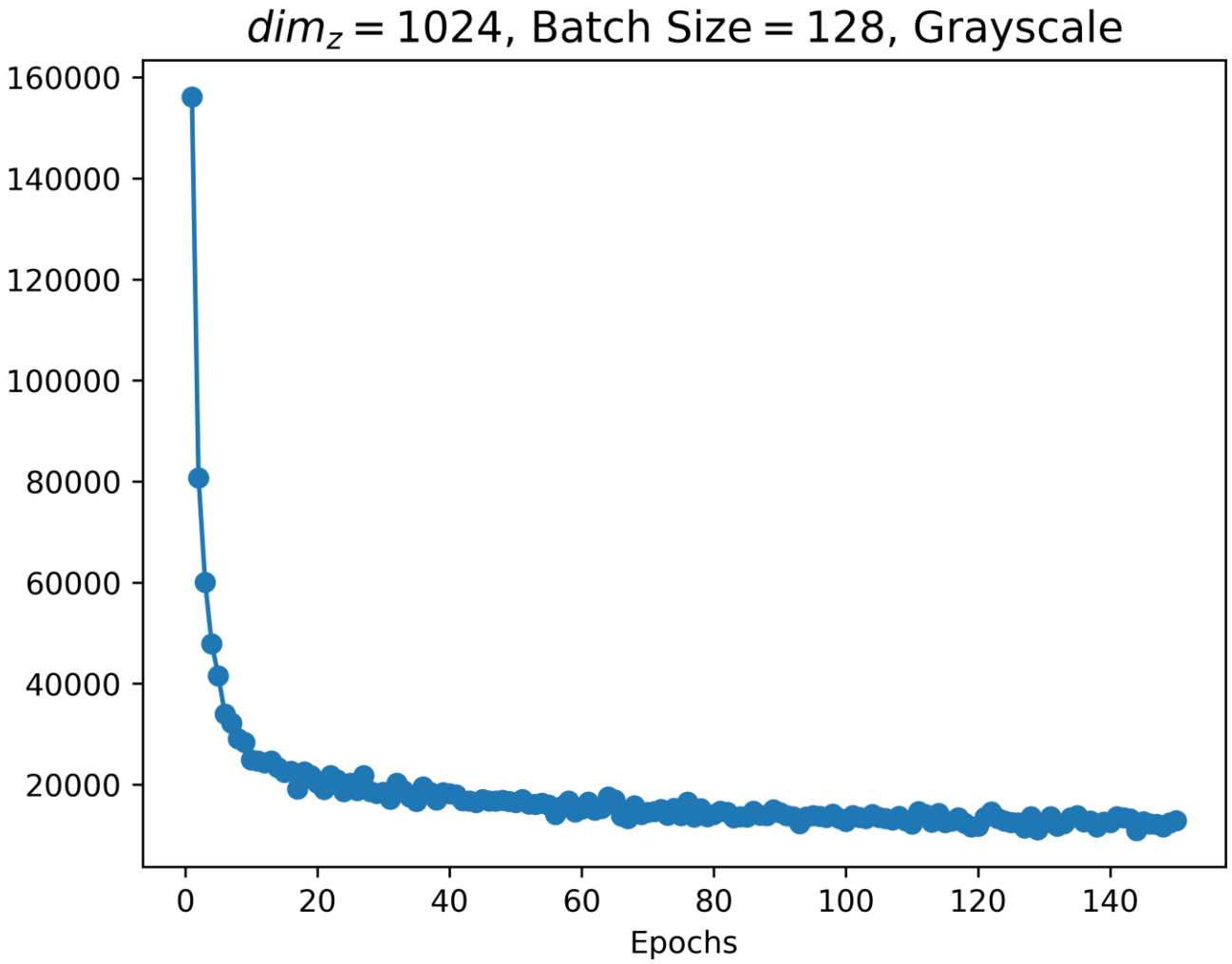




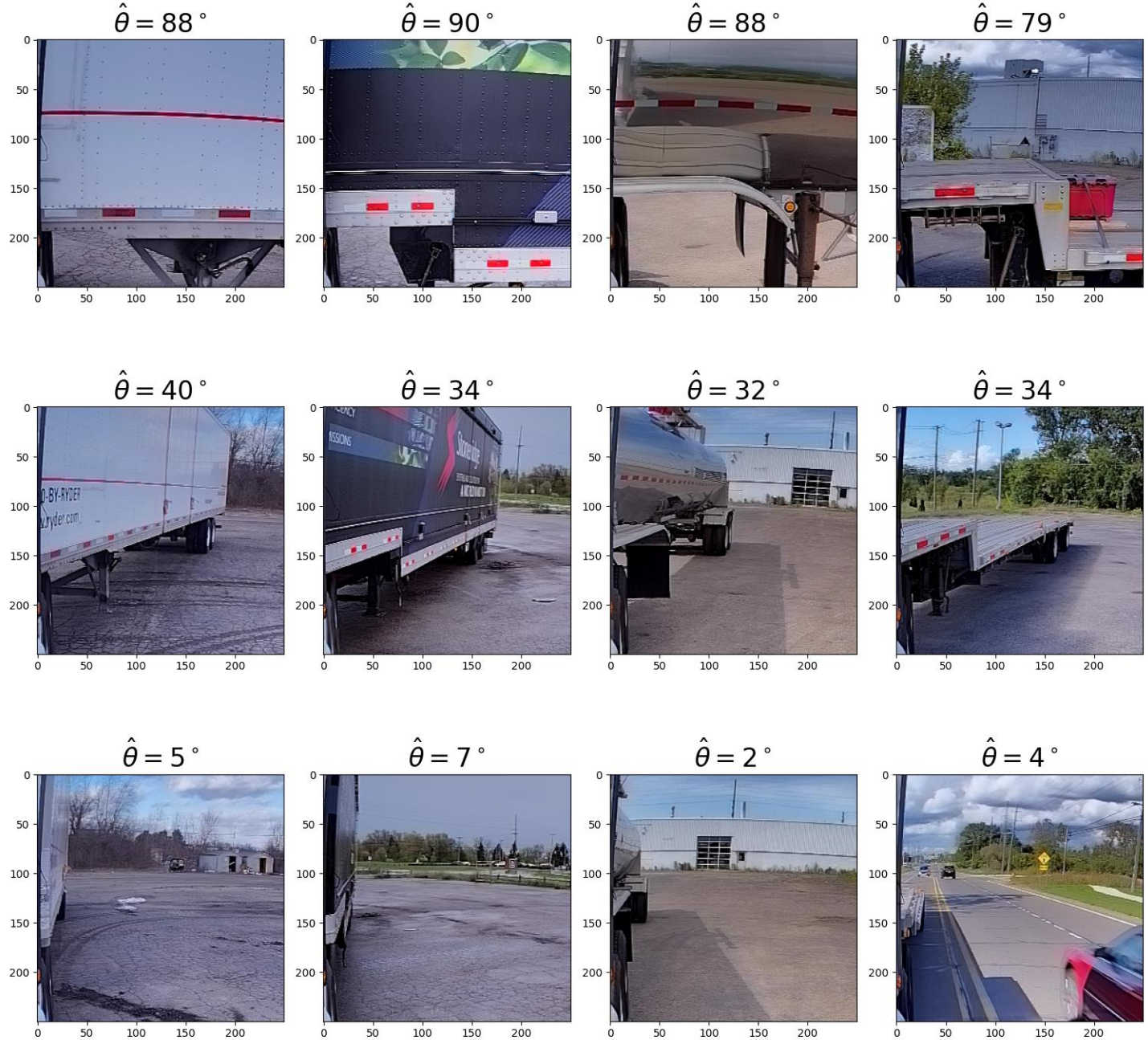
Q/A

Appendix

Appendix


$dim_z = 1024$, Batch Size = 128, Grayscale
160000
140000
120000
100000
80000
60000
40000
20000
0
20
40
60
80
100
120
140
Epochs

BATCH SIZE: 128, β: 50, dim_z: 1024, EPOCHS: 50



BATCH SIZE: 128, β: 50, dim_z: 1024, EPOCHS: 50

BATCH SIZE: 128, β: 50, dim_z: 1024, EPOCHS: 50