CSCI 1430 Final Project Report: Single-Camera Vehicle Trajectory Forecasting on nuScenes Mini

Team name: Charles Tang, Ely Brayboy
TA name: Harper Austin Brown University

Abstract

We built a trajectory forecasting baseline that predicts the next 12 planar waypoints of a target vehicle using only a short history of front-camera frames and recent XY motion history on nuScenes mini. Our model combines a frozen pretrained ResNet18 visual encoder, an LSTM over visual features, an LSTM over motion history, and a small fusion/decoder MLP. We train with mean squared error on normalized trajectories and evaluate with ADE/FDE. In our training run, validation loss reaches its best value near epoch 30, and we observe the expected train–val gap in ADE/FDE over time. We also produce qualitative overlays and scene-diverse examples to analyze strengths and failure cases.

1. Introduction

Vehicle trajectory forecasting is a core perception/planning support problem for autonomous driving. The task is to estimate where an agent (here, a vehicle instance) will move in the near future from scene context and motion history. Even short-horizon prediction is difficult because motion is multi-modal, scenes are cluttered, and camera perspective does not directly encode metric world coordinates.

Our project goal was to build a clean, reproducible baseline for this task on nuScenes mini [1]. To keep scope manageable, we intentionally restricted inputs to a single camera stream (`CAM_FRONT`) plus past XY trajectory history. This lets us study a simple but meaningful visual+motion fusion model before attempting multi-view or full sensor fusion.

If solved robustly, this kind of model can improve downstream motion planning by giving better priors for nearby actor movement in urban driving scenes. In practice, this can support safer lane changes, better braking behavior, and more stable planning in busy intersections where nearby vehicle behavior changes quickly.

2. Related Work

Our data source is nuScenes [1], which provides synchronized multimodal data and 3D annotations in real urban scenes. We used the mini split to iterate quickly while preserving realistic scene diversity.

For visual encoding, we use a pretrained ResNet18 backbone [2]. The intuition is that generic pretrained image features provide strong low-level and mid-level semantics even for a small downstream dataset.

For temporal aggregation, we use LSTMs [3] in both the visual branch and motion branch. Although transformers are currently popular, LSTMs are lightweight and worked reliably for our compute budget and timeline.

Implementation-wise, we rely on PyTorch and torchvision APIs for training and model components.

3. Method

We frame forecasting as supervised regression from image+motion history to future XY positions. Let $\mathbf{I}_{t-k:t}$ denote the last k=5 front-camera frames and $\mathbf{X}_{t-k:t} \in \mathbb{R}^{5\times 2}$ the past XY history in world coordinates. The model predicts $\hat{\mathbf{Y}}_{t+1:t+12} \in \mathbb{R}^{12\times 2}$.

Training loss is MSE on normalized coordinates:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{12}\sum_{\tau=1}^{12} \|\hat{\mathbf{y}}_{t+\tau} - \mathbf{y}_{t+\tau}\|_2^2 , \tag{1}$$

where trajectories are divided by a scale factor of 50 during training (see `code/data.py`).

Evaluation uses ADE and FDE:

$$\text{ADE} = \frac{1}{12}\sum_{\tau=1}^{12} \|\hat{\mathbf{y}}_{t+\tau} - \mathbf{y}_{t+\tau}\|_2 , \tag{2}$$

$$\text{FDE} = \|\hat{\mathbf{y}}_{t+12} - \mathbf{y}_{t+12}\|_2 . \tag{3}$$

Architecture. The final model is:

- Frozen ResNet18 image encoder (per frame, 512-D feature).
- Visual temporal LSTM (hidden size 128).
- Motion LSTM over XY history (hidden size 128).

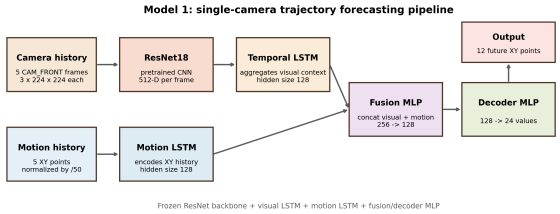

Figure 1. Our single-camera forecasting pipeline used in training and evaluation.

- Fusion MLP on concatenated visual+motion embeddings.
- Decoder MLP outputting 12×2 future points.

Why this model choice. We iterated from a simple design principle: first combine strong pretrained visual features with a lightweight temporal model and explicit motion history, then only increase complexity if needed. This led us to a two-branch design (visual LSTM + motion LSTM) with late fusion. The visual branch captures scene semantics (road context, nearby vehicles, lane geometry cues), while the motion branch captures short-term kinematics that are often predictive even when appearance is ambiguous.

We chose LSTMs instead of heavier temporal models because they were easier to train and debug in our time budget and hardware constraints, and gave stable convergence on mini. We also kept the decoder as an MLP to make the entire model easy to inspect and modify.

Freezing strategy. We freeze the ResNet18 backbone and only train temporal/fusion/decoder heads. This reduced overfitting risk on the small mini split and reduced optimization instability. In our run, this left 11,176,512 parameters frozen and 16 trainable parameter tensors. A likely next ablation is partial unfreezing (e.g., last ResNet block only) after a warmup period.

Data ingestion pipeline. Our pipeline in `code/data.py` does the following:

- Deterministic scene-level splitting (8 train scenes, 2 val scenes) to avoid train/val leakage.
- Per-sample extraction of `CAM_FRONT` frame history (5 frames), with zero-padding on missing history.
- Trajectory history extraction (5 XY points) and future target extraction (12 XY points), with fallback to last known position if an annotation is missing.
- Image preprocessing: resize to 224×224, ImageNet normalization.
- Trajectory normalization by dividing coordinates by 50.0 for stable regression scale.

Hyperparameters and training setup. From `code/train.py`:

- Epochs: 50, early stopping patience: 5 validations.
- Batch size: 4, gradient accumulation: 8 (effective batch size 32).
- Optimizer: Adam, learning rate 1×10^{-3}, weight decay 1×10^{-5}.
- LR schedule: cosine annealing to 1×10^{-5}.
- Gradient clipping: norm 1.0.
- Validation frequency: every epoch.
- Forecast horizon: 12 steps; history length: 5 steps.

Dataset details. For nuScenes mini, our deterministic scene-level split uses 8 train scenes and 2 val scenes. The dataset loader reports 9850 train samples and 1805 val samples, with images resized to 224×224 and ImageNet normalization.

Beyond sample counts, our exploratory statistics show a useful but still limited setting: core mini tables include 10 scenes and 404 samples, with substantial class imbalance across vehicle categories. This imbalance likely contributes to failure cases in rarer motion patterns. We generated supporting summary plots in our poster assets (table counts, top categories, and vehicle class distribution) to guide this analysis.

Compute resources (OSCAR HPC). Most training was run on Brown's OSCAR cluster. For the main logged run (job 1944064), Slurm reports node `gpu2108`, one visible GPU (`CUDA_VISIBLE_DEVICES=0`), 40 GB node memory allocation, and 8 data loader workers during training. This HPC setup was important for finishing 50-epoch experiments and producing multiple evaluation/visualization reruns in a reasonable timeframe.

4. Results

Figure 2 shows train/val loss over epochs. Figure 3 shows ADE/FDE trends over training. In our run, the best validation loss is reached around epoch 30, and metrics generally improve early then flatten, which is consistent with diminishing returns on the mini split.

Qualitatively, performance is best on steady-motion scenarios and moderate-curvature trajectories where recent motion is a strong predictor. Harder cases include abrupt turns, interaction-heavy scenes, and frames where the target vehicle is partially off-camera or visually small. This behavior is expected for a single-camera baseline without explicit multi-agent interaction modeling.

To better analyze generalizability, Figure 4 shows examples from multiple train and validation scenes rather than a

Metric (best val epoch)	Value
Val Loss	0.0495
Val ADE	1.2203 m
Val FDE	1.9169 m

Table 1. Best validation metrics from the logged training run.

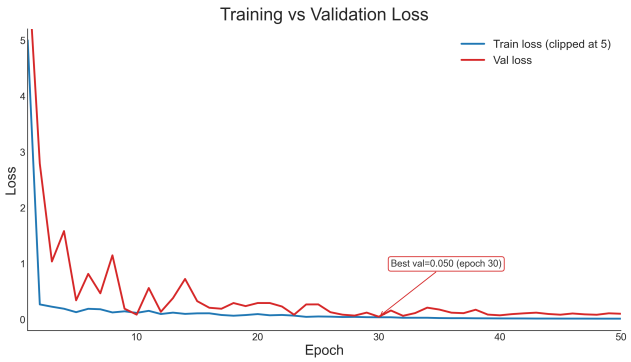

Figure 2. Train and validation loss across epochs.

single cherry-picked frame. Across these scenes, the model usually captures heading and short-horizon drift correctly when traffic flow is smooth, but it degrades in sharper maneuvers and more interaction-dense settings. This supports the claim that our baseline learns transferable short-horizon motion cues, while still lacking robustness to rare or highly multimodal behavior.

Brief error analysis. Looking across our rendered scene set, we repeatedly observed three practical error modes: (1) *turn underestimation*, where predicted trajectories remain too straight during sharper lane changes or intersection turns; (2) *interaction lag*, where predictions react too slowly when nearby vehicles induce sudden speed or heading changes; and (3) *endpoint drift*, where final predicted points overshoot in depth when the target is small or partially occluded. These error modes are consistent with a single-camera, short-history setup and motivate stronger temporal context and interaction-aware modeling.

4.1. Technical Discussion

The model is intentionally simple, which gave us a strong baseline quickly, but also introduced clear trade-offs:

- **Pros:** stable training, interpretable architecture, low engineering complexity, and fast iteration on mini.
- **Cons:** no multi-view geometry, limited context for occlusions/turns, and likely underfitting of scene semantics with frozen visual backbone.

One key practical lesson is that qualitative inspection matters. We added overlay renderings directly on camera frames to inspect where predictions look plausible versus where they drift. This helped us find failure cases where trajectories are off-camera or the motion mode is ambiguous.

Another useful lesson is that training stability and data quality checks mattered as much as architecture. Gradient accumulation, clipping, and deterministic scene splits reduced noisy behavior in training and made metric trends easier to interpret.

4.2. Ablation Plan

To make our conclusions more evidence-driven, we propose the following ablations as immediate next experiments:

- **Vision-only vs motion-only vs fused model:** quantify how much each branch contributes to ADE/FDE.
- **Freeze vs partial unfreeze:** compare frozen ResNet18 to unfreezing only the last residual block after warmup.
- **Temporal horizon sensitivity:** test history lengths (3, 5, 8) and forecast horizons (6, 12) to measure stability vs long-range drift.
- **Capacity sweep:** increase LSTM hidden size and decoder width to check whether current errors are under-capacity vs data-limited.

For each ablation, we would report ADE/FDE plus qualitative overlays on the same fixed scene subset to keep comparisons fair and interpretable.

4.3. Autonomous Driving Relevance

Even though this project is a class-scale baseline, the setup mirrors real planning support pipelines: predict near-future actor motion from perception outputs, then pass those forecasts to a planner. In autonomous driving, ADE/FDE improvements can translate into better safety margins and fewer planner oscillations, especially in stop-and-go traffic and intersections. Our current model is not deployment-ready, but it provides a practical scaffold for integrating richer context and uncertainty modeling.

4.4. Future Extension: Full Trainval Split

Our most direct next step is to scale from mini to full `v1.0-trainval` (700 scenes). We already support this in our scripts by setting `NUSCENES_VERSION=v1.0-trainval`. We expect three concrete benefits from this extension:

- Better generalization from broader scene/weather/traffic diversity.
- Reduced overfitting pressure, enabling more aggressive fine-tuning (e.g., unfreezing more of the visual backbone).

Figure 3. *Left:* ADE/FDE curves over training. *Right:* Qualitative trajectory overlay example (GT vs prediction) from scene-diverse rendered outputs.

- More meaningful model comparisons and ablations due to larger validation coverage.

Expected challenges are significantly longer training times, larger storage/IO costs, and potentially needing more careful sampling to balance common versus rare behaviors.

Compute and data-access limitations. In practice, scaling is not only a modeling decision but also a systems constraint. Downloading and maintaining full trainval requires substantially more local/cluster storage and longer dataset preparation time than mini, which slows iteration cycles. On the training side, adding many multimodal inputs (multiple cameras plus additional sensor streams) with temporal models increases memory pressure and input-pipeline overhead. With LSTM-based temporal fusion, sequence length and number of modalities both increase activation memory and loader bandwidth demands, making batch-size tuning and epoch time much more expensive. Under our course timeline, this made a focused single-camera baseline the most feasible path to complete, validate, and analyze end-to-end.

4.5. What We Still Need (Useful but Missing)

If we had more time, these would make the report stronger:

- A direct comparison table against at least one motion-only baseline and one stronger published trajectory baseline.
- Confidence intervals across multiple random seeds (currently we report a single run).
- Ablation studies: freeze vs unfreeze backbone, no visual branch, no motion branch, different history/future horizons.
- Runtime throughput and memory profiling (train and eval) under identical hardware settings.
- More rigorous qualitative taxonomy (e.g., turns, occlusions, intersections, dense traffic).

5. Conclusion

We implemented and evaluated a single-camera trajectory forecasting baseline on nuScenes mini that fuses visual context and motion history. The model reaches reasonable validation loss/ADE/FDE for the constrained setting and produces interpretable trajectory overlays across multiple train/val scenes. Our results suggest that short-horizon heading and drift are learned reliably in common driving patterns, while complex interaction-heavy and high-curvature cases remain challenging.

A key outcome of this project is the complete experimental pipeline we built: deterministic data ingestion, stable training with checkpoint selection, metric tracking (loss/ADE/FDE), and qualitative rendering for failure inspection. This pipeline is already useful beyond the current model because it supports reproducible iteration and fair comparison of future variants.

At the same time, we found clear system-level constraints that shaped design choices. Full trainval and heavily multimodal temporal fusion are promising for generalization,

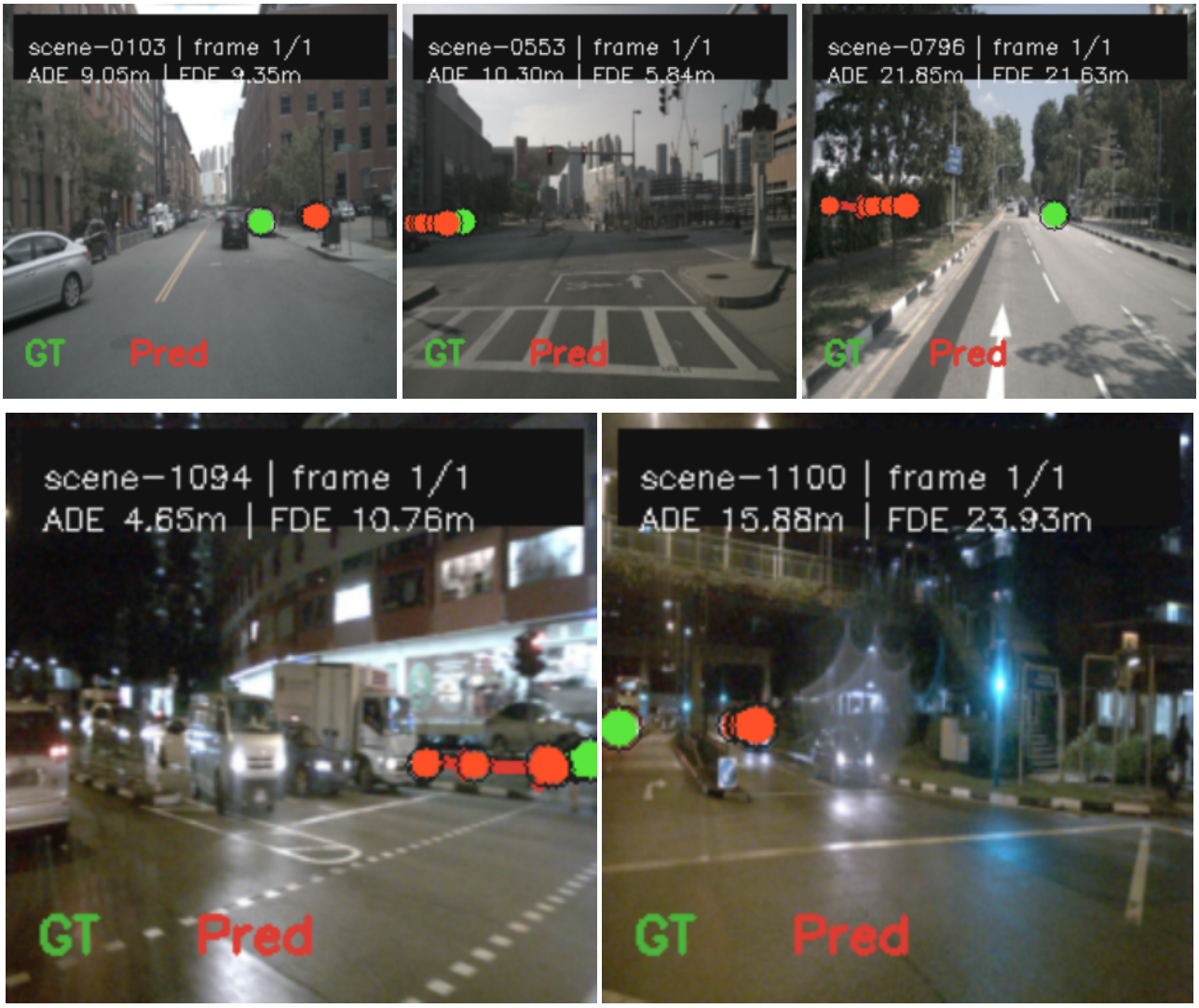

Figure 4. Multi-scene qualitative overlays for generalizability analysis. Top row: train scenes. Bottom row: validation scenes. The model is reasonably consistent on common driving patterns across scenes, but larger endpoint errors appear in more complex motion contexts.

but they are constrained by storage, data-loading bandwidth, and training-time budget in a course setting. Our immediate next step is therefore structured ablation (branch contribution, freezing strategy, temporal horizon) before scaling up model/sensor complexity. This staged approach should provide stronger evidence for which changes materially improve generalization and which only increase compute cost.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1

[3] Sepp Hochreiter and J"urgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 1

Appendix

Team contributions

Charles Tang I led model training/debugging and most of the experiment operations on OSCAR (job setup, checkpoint management, log collection, and reruns). I also worked on evaluation visualization quality, including trajectory overlays and scene-diverse qualitative examples used in the poster/report.

Ely Brayboy I focused on project framing, dataset analy-

sis, architecture communication, and report/presentation materials. I contributed to design decisions around the baseline scope (single-camera + motion history), interpretation of results, and final narrative for methods/results sections.