

# **Model Fusion: Weighted N-Version Programming for Resilient Autonomous Vehicle Steering Control**

**Ailec Wu\*, Abu Hasnat Mohammad Rubaiyat, Chris Anton, Homa Alemzadeh**

**Electrical and Computer Engineering**

**University of Virginia**



**ENGINEERING**

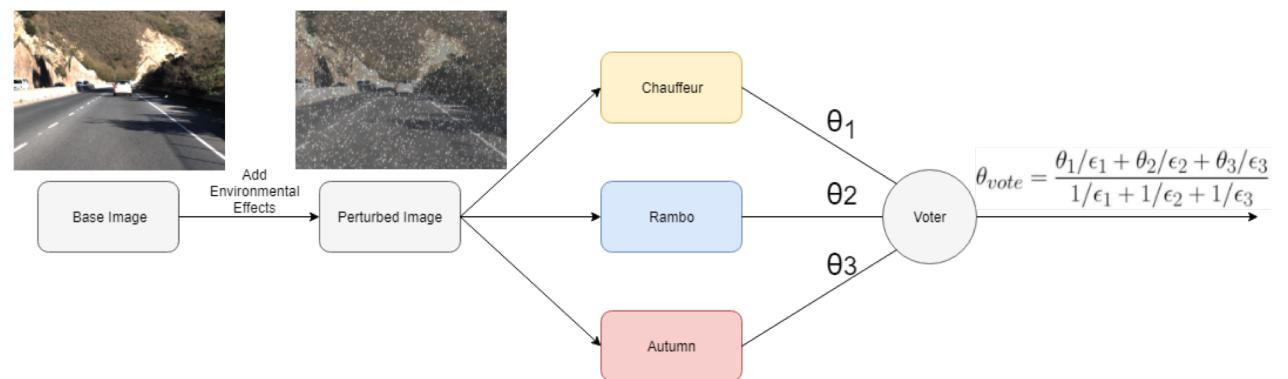
# Motivation



Individual Models vs. Ground Truth: Rain, Snow

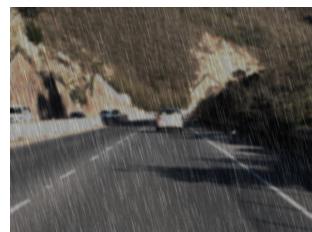
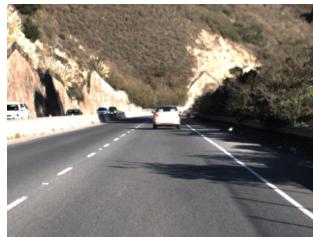
# Objective

- Improve neural network based lane keep assistance systems resilience
  - Perturbations caused by sensor faults or environmental conditions
- Use N version programming inspired approach
  - Three models from the Udacity self-driving car challenge, independently designed by separate teams
  - Weighted fusion of output for voting
- Performance evaluation:
  - Compare accuracy of combined NVP system vs. individual models



# Evaluation

- Fault injection to images to mimic real world perturbations/environmental conditions
  - Structural Similarity (SSIM) index <0.6
- Performance of individual models vs. NVP system evaluated
  - Root Mean Square Error (RMSE)



Rain



Snow



Fog



Brightness

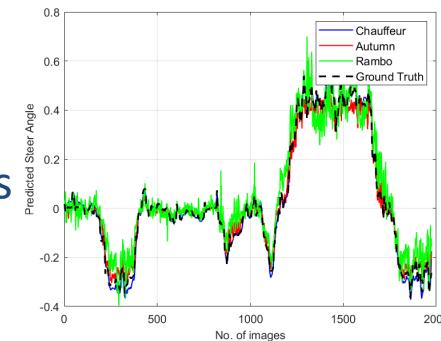


Contrast

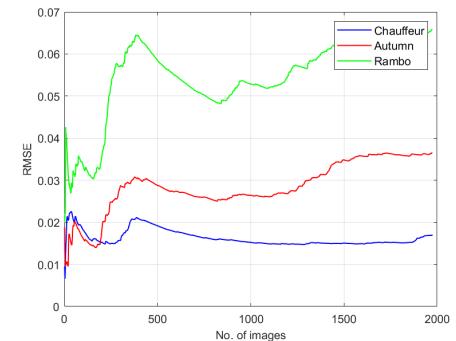
# Reliability of Individual Models

- Three pre-trained models from Udacity's self-driving car challenge
  - Chauffer:** one CNN model for feature extraction and one LSTM/RNN model for predicting steering angles
  - Rambo:** three CNNs whose outputs are merged via a final layer
  - Autumn:** five CNNs and an LSTM/RNN layer

Algorithm	RMSE
Chauffeur	0.0163
Autumn	0.0285
Rambo	0.0548



Predicted Steering Angle

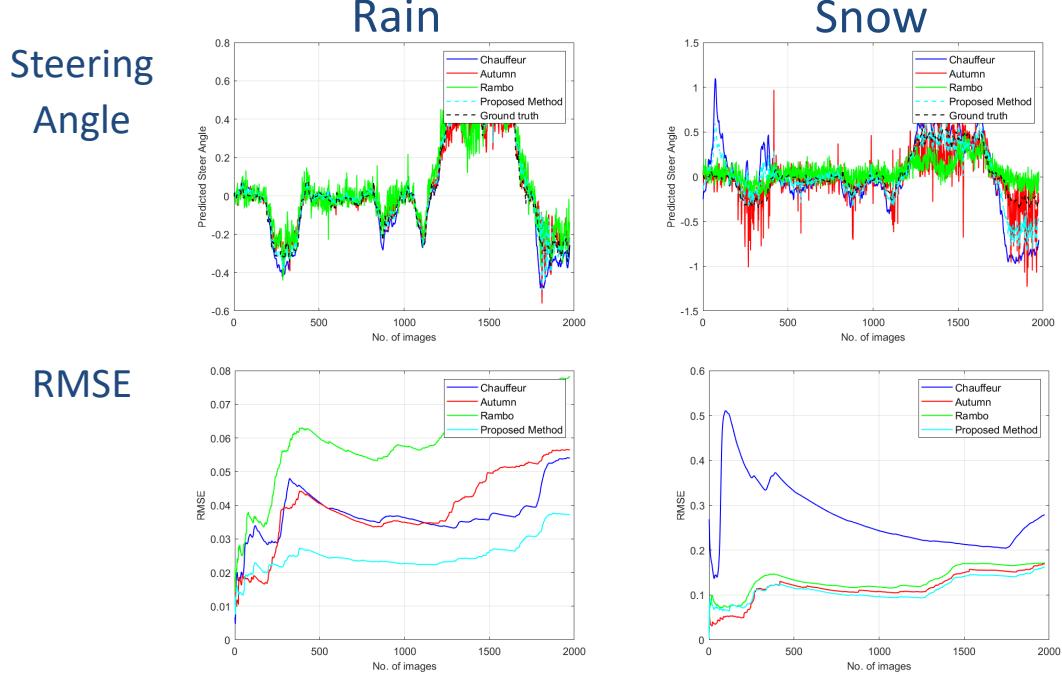


Root Mean Squared Error

# Weighted Averaging

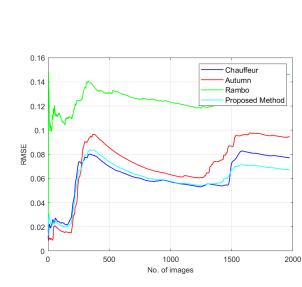
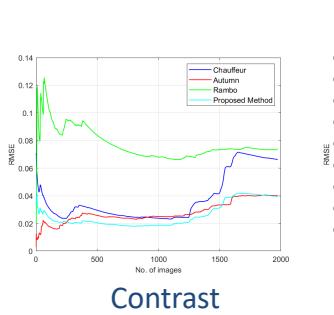
- N-Version Programming needs to account for:
  - Some models are more reliable than others
  - Variables are continuous; models have error
- Weighted averaging handles asymmetry
  - Weight inversely proportional to RMSE

$$\theta_{vote} = \frac{\theta_1/\epsilon_1 + \theta_2/\epsilon_2 + \theta_3/\epsilon_3}{1/\epsilon_1 + 1/\epsilon_2 + 1/\epsilon_3}$$



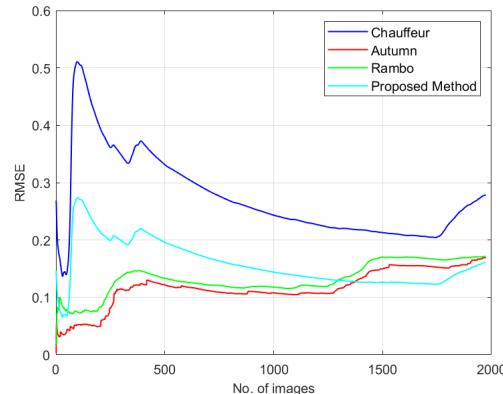
# Weighted Averaging + Exclusion

- Exclude the faulty model
  - Examine differences in model outputs
  - Still need to account for outliers
  - Exclude models with differences above noise ceiling

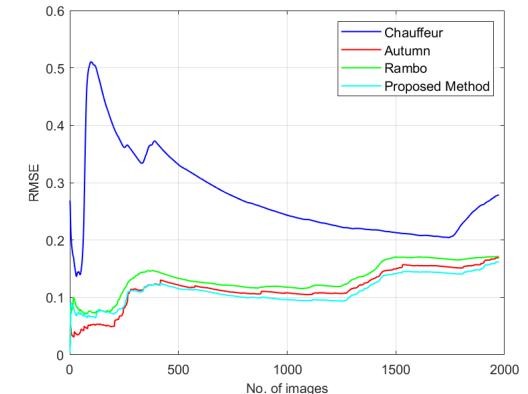


$$\Delta\theta_{12} \approx \Delta\theta_{13} \not\approx \Delta\theta_{23} \implies \theta_1 \not\approx \theta_2 \approx \theta_3$$

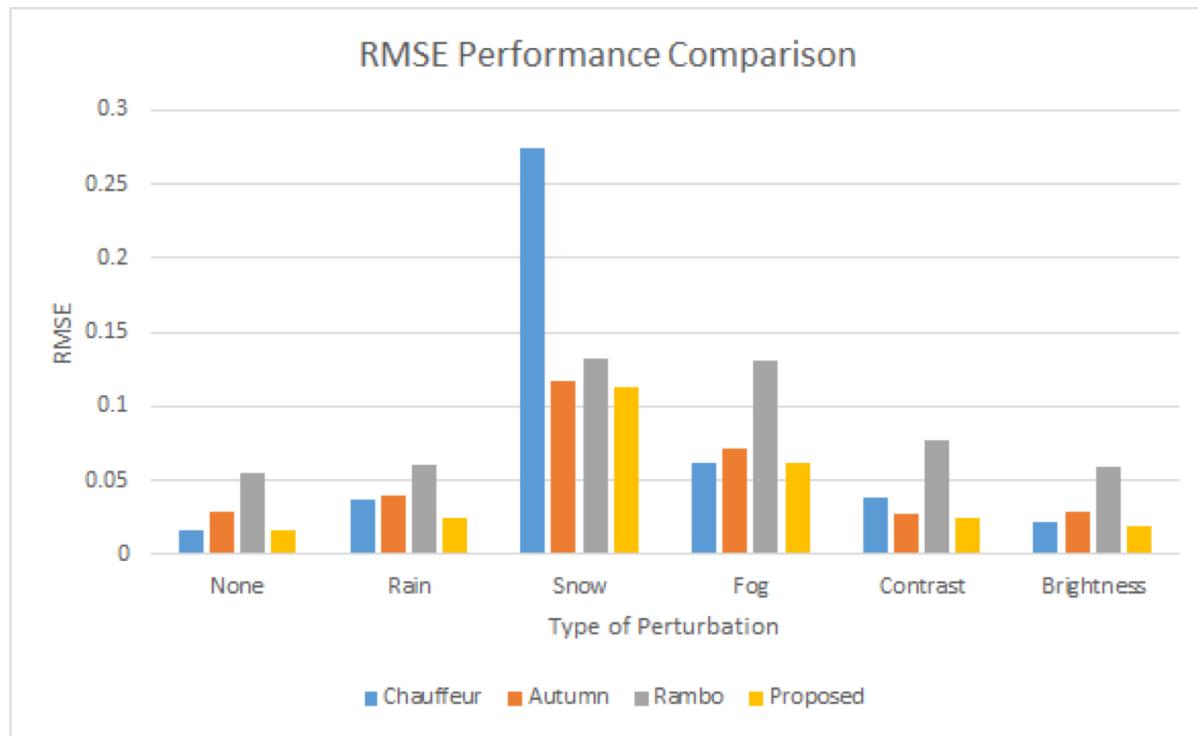
Check differences and exclude outliers



Weighted Averaging Only



Weighted NVP



# Demonstration



Proposed System vs. Ground Truth: Rain, Snow

# Future Work

- **Reliability Analysis:**
  - Evaluate the impact of number of models, model independence, and failure rates on reliability
    - Using diverse training datasets and model designs
- **Cost of NVP:**
  - Not only developing independent models, but also diverse dataset creation and model training time
- **Exclusion Logic:**
  - Extend the exclusion logic to detect faulty model(s) when having a higher number of versions and different noise levels