Supplementary Material for Rules of the Road: Predicting Driving Behavior with a Convolutional Model of Semantic Interactions

1. Overview

- The bivariate normal negative log-likelihood loss function to model uncertainty into regression trajectories is

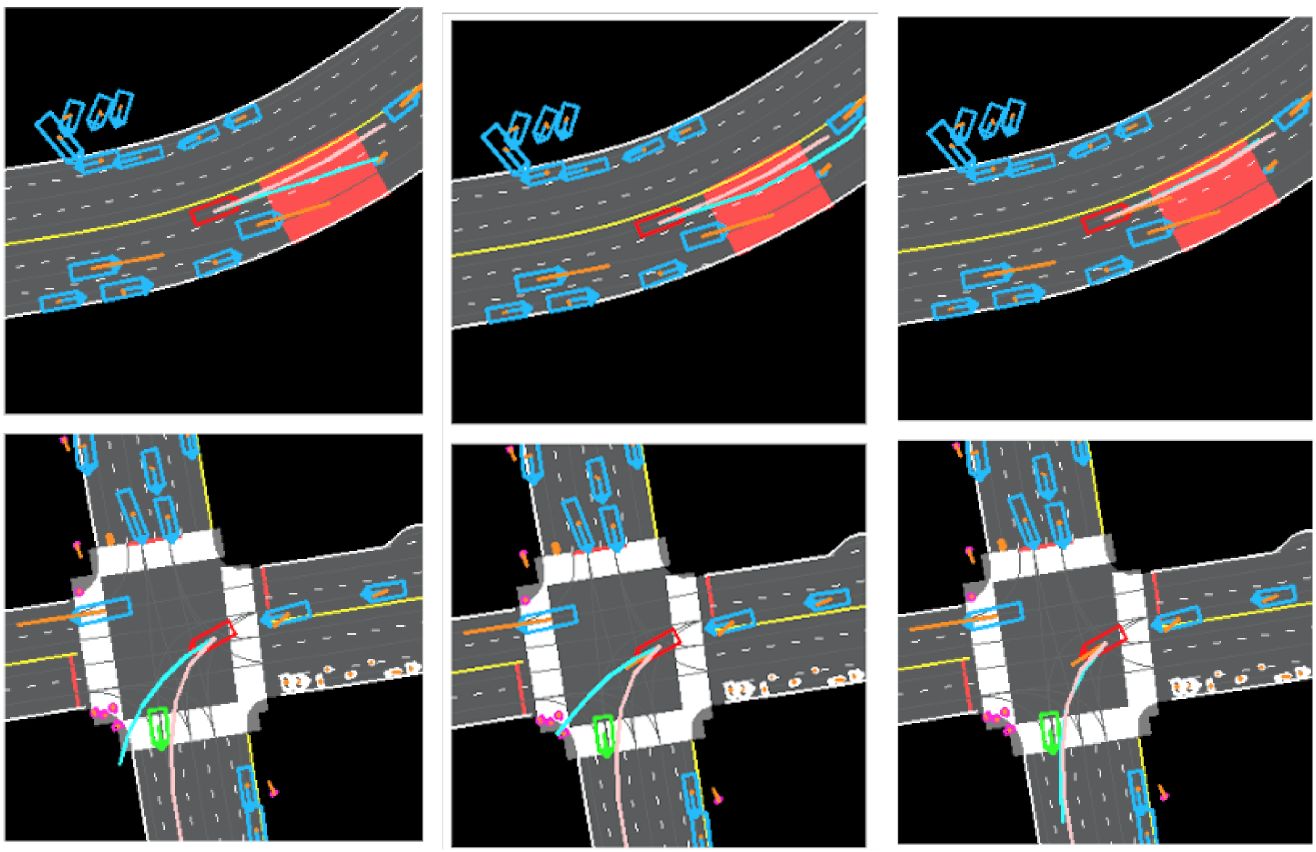
$$
\begin{aligned}
\mathcal{L}(\theta) = \sum_{i=1}^{5} & \log \sigma_{x_i} + \log \sigma_{y_i} + \log \sqrt{1-\rho_i^2} \\
& + \frac{1}{2(1-\rho_i^2)} \left(\frac{x_i - \mu_{x_i}^2}{\sigma_{x_i}^2} + \frac{y_i - \mu_{y_i}^2}{\sigma_{y_i}^2} - \frac{2\rho_i (x_i - \mu_{x_i})(y_i - \mu_{y_i})}{\sigma_{x_i}\sigma_{y_i}} \right).
\end{aligned}
\tag{1}
$$

  Figure 1 shows our Tensorflow implementation of this loss.

- Figure 2 shows examples where adding more scene context improves performance dramatically.
- Figure 3 shows examples where the Grid Map method gives outperforms the others, with heatmap visualization.
- Figure 4 shows more successfuly and unsuccesful examples of the Grid Map method with corresponding heatmap visualizations.

```
def binormal_log_likelihood_loss(y_true, y_pred):
  """
  y_{pred,true} are of dimensions batch x 25. The second dimension holds 5-tuples
  (mu_x, mu_y, log_var_x, log_var_y, rho) for each delta_t = 1,...,5 seconds in the future,
  in the order:
  [mu_x^1, ...,mu_x^5, ..., log_var_x^1, ..., log_var_x^5, ..., rho^1, ..., rho^5],
  with superscript indexing time step.
  """

  mu_pred, log_var_pred, rho = tf.split(y_pred, [10, 10, 5], axis=1)
  errs = tf.reshape(y_true, mu_pred, [-1, 5, 2])
  log_var_pred = tf.reshape(log_var_pred, [-1, 5, 2])

  # Regularize the predicted variances.
  var_reg_loss = 0.5 * tf.reduce_sum(log_var_pred, axis=-1) +
      tf.log(tf.sqrt(1 - tf.square(rho)) + 1e-20)

  # Regression loss terms.
  loss_individual = tf.reduce_sum(tf.multiply(tf.square(errs),
                                      tf.exp(-log_var_pred)),
                                      axis=-1)
  loss_cross = tf.multiply(tf.multiply(rho, tf.reduce_prod(errs, axis=-1)),
                               tf.exp(-0.5 * tf.reduce_sum(log_var_pred, axis=-1)))

  # Combine all the losses to get the bivariate Gaussian loss.
  loss_mat = var_reg_loss + 0.5 * tf.multiply(tf.reciprocal(1 - tf.square(rho) + 1e-20),
                                                  loss_individual - 2 * loss_cross)

  return tf.reduce_mean(loss_mat, axis=-1)
```

Figure 1: TensorFlow code for normal distribution log-likelihood loss

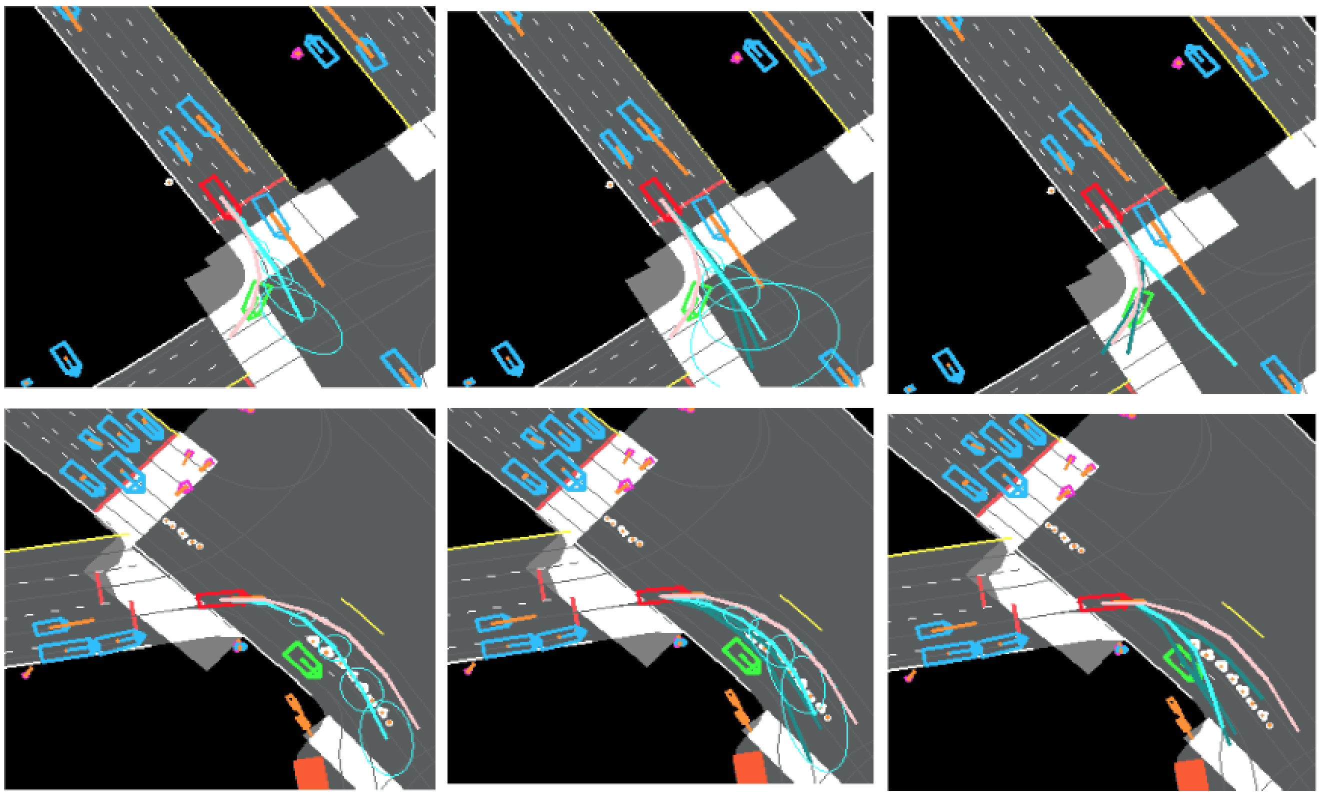


(a) Target

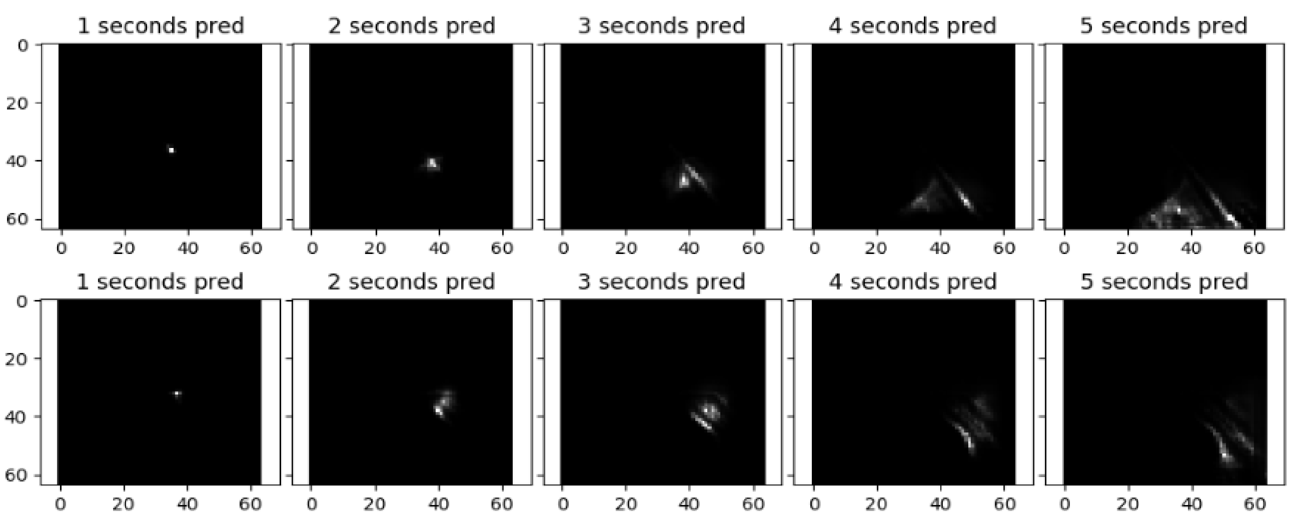
(b) Target+Other

(c) Target+Other+Map

Figure 2: Examples from ablation study using the Gaussian Regression method. In the **top** example, the predicted car trajectory veers off; adding other entities allows the model to learn some following behavior, and the map allows the model to follow lane lines. In the **bottom**, the car is turning at a junction, which the model learns only after including the road map.

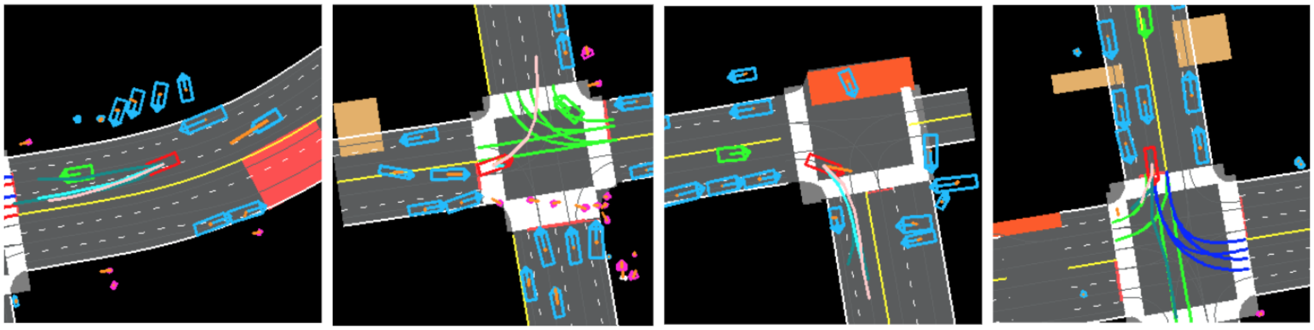


(a) Gaussian Regression (b) GMM-CVAE (c) Grid Map

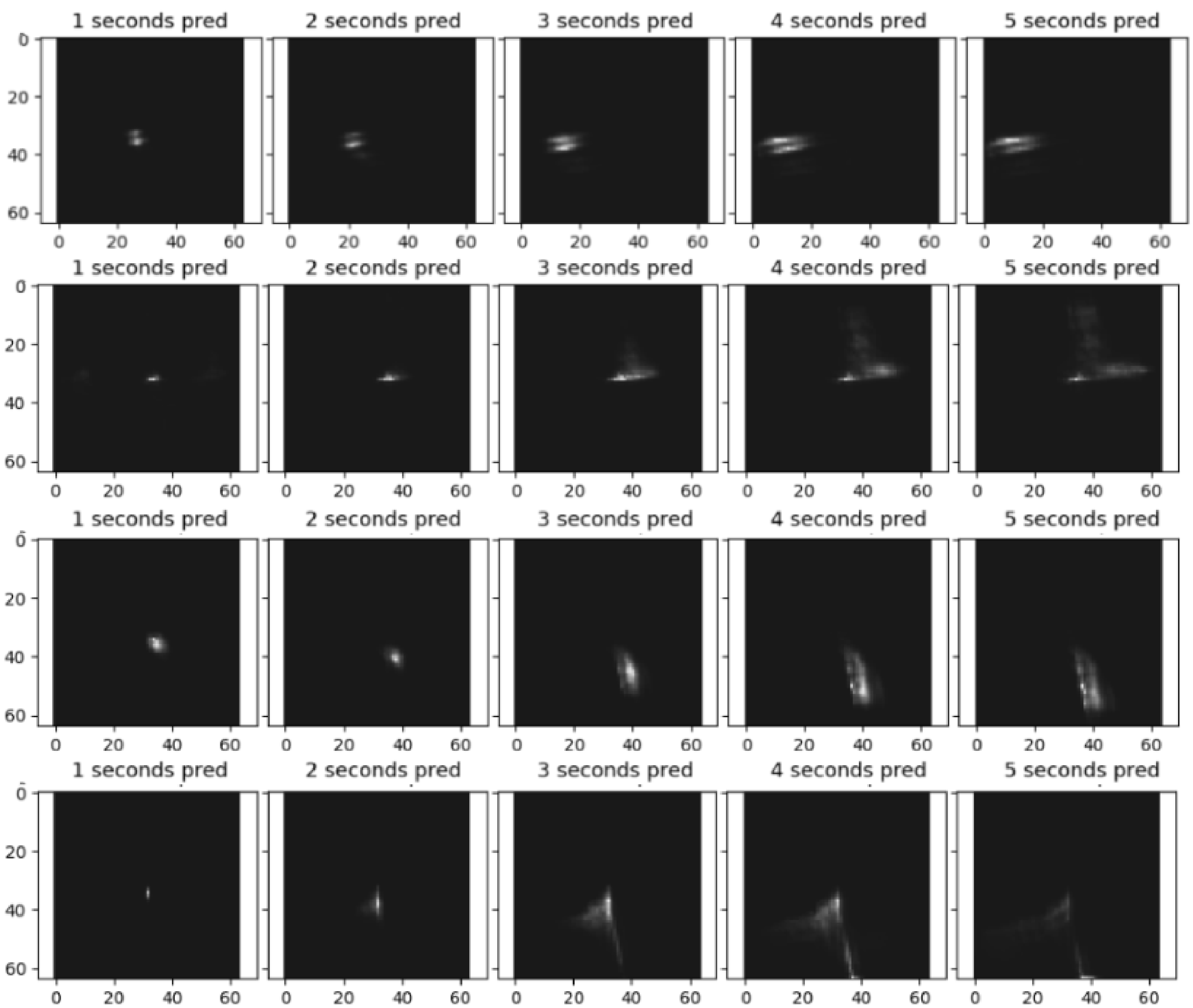


(d) Grid Map visualized.

Figure 3: Cases where the Grid Map approach outperforms the Gaussian approaches, in driving scenarios which feature multiple possible outcomes. In both cases, the Grid Map approach predicts diverse, likely outcomes, and covers the ground truth. In the **top**, the car is entering a junction, and in the **bottom**, the car is approaching a fork with obstacles. The Grid Map method effectively predicts multiple, disparate modes, and trajectories are sampled from both. In (d), the Grid Maps are visualized for the top and bottom examples—note the variable number of modes with non-elliptical uncertainty.



(a) Grid Map



(b) Grid Map visualized.

Figure 4: Some more examples of the Grid Map approach in (a), and their corresponding visualizations in (b). In all the examples the heatmaps show high quality diverse predictions. In the first example, we see that the heatmaps can even show detailed road semantics as lane lines. In the middle two, the visualizations showed multiple, diverse modes, but our sampling procedure did not capture all of them.