Supplementary Material
DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents

Namhoon Lee[1], Wongun Choi[2], Paul Vernaza[2], Christopher B. Choy[3],
Philip H. S. Torr[1], Manmohan Chandraker[2,4]

[1]University of Oxford, [2]NEC Labs America, [3]Stanford University, [4]University of California, San Diego

1. Network Details

In this section, we describe the additional details of the network architecture of *DESIRE*. We summarize the full details in Table. 1 and discuss them in the following subsections.

1.1. Convolutional Neural Networks for Static Scene Context

We use a CNN to encode the static spatial context of the scene. For the KITTI dataset, two convolutional layers (*conv1* and *conv2*) with *ReLU* activation are employed. For the SDD dataset, we add an additional convolution layer. We adopt a relatively shallow CNN for an ease of training, but a deeper network could be adopted with a use of pre-trained network parameters.

1.2. RNN Encoder 1

RNN Encoder 1 is responsible for encoding the past motion of the individual agent. The encoder is implemented with a temporal convolution layer (a.k.a, 1D convolution) that is followed by an RNN with GRU cells [1]. Before putting the past trajectory (X_i) to the temporal convolution layer, we subtract the last state value (present location) from $x_{i,t}$ at all time steps for a translation invariance. The temporal convolution layer has a kernel with 3 frames and 16 channel outputs. The GRU RNN has a 48 dimensional hidden vector, where the initial hidden vector is padded with 0.

1.3. Conditional Variational Auto-encoder

The CVAE module is composed of a recognition network $Q_\phi(z_i|Y_i, X_i)$, a prior network $P_\nu(z_i)$, and a generation network $P_\theta(Y_i|X_i, z_i)$.

- $Q_\phi(z_i|Y_i, X_i)$ is implemented with a neural network that generates a latent variable z_i given the encodings of X_i and Y_i. The output of RNN Encoder 1 ($\mathcal{H}_{X_i}$) is used as the encoding of X_i. We implement a similar network to provide the encoding of Y_i through RNN encoder 2 that outputs a 48 dimensional encoding ($\mathcal{H}_{Y_i}$). Similarly, we use a temporal convolution layer and GRU with the parameters specified in Table. 1. $\mathcal{H}_{X_i}$ and $\mathcal{H}_{Y_i}$ are concatenated and passed through three *fully connected* layers to produce μ_i and σ_i. We sample 48 dimensional $z_i^{(k)}$ using the *reparameterization trick* [2], i.e., $z_i^{(k)} = \mu_i + \sigma_i \boxtimes \epsilon_i^{(k)}$, $\epsilon_i^{(k)} \sim \mathcal{N}(0, 1)$. During the testing phase, we randomly draw $z_i^{(k)}$ from the prior distribution $P_\nu(z_i) := \mathcal{N}(0, 1)$.
- $P_\theta(Y_i|X_i, z_i)$ is implemented with another GRU-RNN (RNN decoder 1) that takes both $\mathcal{H}_{X_i}$ and $z_i^{(k)}$ as inputs. The two inputs are mixed together as discussed in the paper. The mixed vector $xz_i^{(k)}$ is provided as the input of the RNN decoder 1 at initial time frame, and all the other inputs at later time steps are padded with 0. The initial hidden vector of the RNN is initialized with 0. The RNN decoder 1 is implemented with a GRU with 48 dimensional hidden vector that is followed by a *fully connected* (fc) layer that produces 2 dimensional state reconstruction at every time frame. Instead of directly reconstructing the absolute state (location), we estimate the displacement relative to the previous time step and add them to the pervious state. We share the parameter of the fc layer over all time steps.

Layer Type	Input (dimensions)	Output (dimensions)	Additional Parameters
CNN $\rho(I)$			
2D-Convolution	$I, (H, W, 4)$	$conv1, (H/2, W/2, 16)$	act:=ReLU, kernel:= $(5, 5)$, stride:=2
2D-Convolution	$conv1, (H/2, W/2, 16)$	$conv2, (H/2, W/2, 32)$	act:=ReLU, kernel:= $(5, 5)$, stride:=1
2D-Convolution	$conv2, (H/2, W/2, 32)$	$conv3, (H/2, W/2, 32)$	act:=ReLU, kernel:= $(5, 5)$, stride:=1
RNN Encoder 1			
1D-Convolution	$X_i, (2, 20)$	$tX_i, (16, 20)$	kernel:=(3), act:=ReLU
GRU	$tX_i\ (16, 20)$	$\mathcal{H}_{X_i}\ (48)$	RNN length:=20
Conditional Variational Auto-encoder			
RNN Encoder 2 (training phase)			
1D-Convolution	$Y_i, (2, 40)$	$tY_i, (16, 40)$	kernel:=(1), act:=ReLU
GRU	$tY_i, (16, 40)$	$\mathcal{H}_{Y_i}, (48)$	RNN length:=40
Q distribution (training phase)			
Concat.	$\mathcal{H}_{X_i}, \mathcal{H}_{Y_i}, (48), (48)$	$\mathcal{H}_{XY_i}, 96$	-
Fully-connected	$\mathcal{H}_{XY_i}, (96)$	$fc^1_i, (48)$	act:=ReLU
Fully-connected	$fc^1_i, (48)$	$fc^\mu_i, (48)$	act:=Linear
Fully-connected	$fc^1_i, (48)$	$fc^\sigma_i, (48)$	act:=$\frac{1}{2}\exp(\cdot)$
Reparam. trick	$fc^\mu_i, fc^\sigma_i, (48), (48)$	$z^{(k)}_i, 48$	$z^{(k)}_i = \mu_i + \sigma_i \boxtimes \epsilon^{(k)}_i, \epsilon^{(k)}_i \sim \mathcal{N}(0, 1)$
Q distribution (testing phase)			
Random.	-	$z^{(k)}_i, (48)$	$z^{(k)}_i \sim \mathcal{N}(0, 1)$
Sample Reconstruction			
Fully-connected	$z_i, (48)$	$fc^z_i, (48)$	act:=Linear
Softmax	$fc^{z_i}, (48)$	$\beta(z_i), (48)$	-
Multiplication	$H_{X_i}, \beta(z_i), (48), (48)$	$xz^{(k)}_i$	$\mathcal{H}_{X_i} \boxtimes \beta(z^{(k)}_i)$
GRU	$xz^{(k)}_i, (48)$	$hxz^{(k)}_{i,t}, (48) \times 40$	RNN length:=40
Fully-connected	$hxz^{(k)}_{i,t}, (48)$	$\hat{y}^{(k)}_{i,t}, (2)$	act:=Linear
IOC Ranking and Refinement			
Scene Context Fusion			
Feature-pooling	$\rho(\mathcal{I}), \hat{y}^{(k)}_{i,t}, (H/2, W/2, 32), (2)$	$p^{(k)}_{i,t}, (32)$	Pool the features at $\hat{y}^{(k)}_{i,t}$
Fully-connected	$\hat{v}^{(k)}_{i,t}, (2)$	$fv^{(k)}_{i,t}, (16)$	act:= ReLU
Social-pooling	$\hat{y}^{(k)}_{i,t-1}, \hat{y}^{(l)}_{j,t-1}, h^{(l)}_{j,t-i} \forall j \neq i, \forall l$	$sp^{(k)}_{i,t}, (6 \times 6 \times 48)$	pool := Average
Fully-connected	$sp^{(k)}_{i,t}, (6 \times 6 \times 48)$	$fsp^{(k)}_{i,t}, (48)$	act:= ReLU
Concat.	$p^{(k)}_{i,t}, fv^{(k)}_{i,t}, fsp^{(k)}_{i,t}, (32), (16), (48)$	$scf^{(k)}_{i,t}, (96)$	-
RNN Decoder 2			
GRU	$scf^{(k)}_{i,t}, (96) \times 40$	$h^{(k)}_{i,t}, (48) \times 40$	RNN length:=40, $h^{(k)}_{i,0} = \mathcal{H}_{X_i} \forall k$
Fully-connected	$h^{(k)}_{i,t}, (48)$	$\psi^{(k)}_{i,t}, (1)$	act:= Linear
Fully-connected	$h^{(k)}_{i,T}, (48)$	$\triangle\hat{Y}^{(k)}_i, (2, 40)$	act:= ReLU

Table 1. Detailed architecture of *DESIRE*.

1.4. IOC Ranking and Refinement

The IOC ranking and refinement module is implemented with an RNN that takes the outputs of the Scene Context Fusion (SCF) unit as an input at each time step. The past motion context $\mathcal{H}_{X_i}$ is provided as the initial hidden vector of the RNN. The RNN is implemented with 48 dimensional GRU that is followed by a fc layer that produces one dimensional score output at each time step and another fc layer that yields the 2×40 dimensional regression vector $\triangle\hat{Y}^{(k)}_i$ at the last time step.

Scene Context Fusion: The SCF unit combines velocity of prediction sample at each time step, surrounding static scene context through the feature pooled from the CNN, and dynamic scene context through our social pooling layer. The velocity feature is obtained by passing the raw velocity $\hat{v}^{(k)}_{i,t}$ at each time step through a fc layer with 16 dimensional outputs and *ReLU* activation. The static scene feature is obtained by pooling the 32 dimensional feature vector from the corresponding location ($\hat{y}^{(k)}_{i,t}$) of the last convolution layer of the CNN (*conv2* for KITTI and *conv3* for SDD). The social pooling layer aggregates the contextual information about how the other agents are moving with respect to an agent. We implement the layer with a log-polar grid which has 6 radial bins and 6 angular bins. The radial bins are discretized in a log space with minimum and maximum distance (d_{min}, d_{max}). We use $(0.5\ m, 4\ m)$ for the KITTI dataset and $(1\ pixel, 40\ pixel)$ for the SDD dataset. The hidden vectors of each prediction sample are aggregated though an average pooling operation within a grid. The three vectors are concatenated and provided as an input of the GRU at each time step.

2. Additional Qualitative Examples

2.1. Qualitative Examples for KITTI Results

DESIRE considers potential long-term future rewards formulated as an IOC framework. Thus, DESIRE produces more accurate prediction than other baselines such as RNN Encoder-Decoders (with or without SCF) which often behave *reactively* (as depicted in Fig. 1 - Row 1, 2, 3, 4). Moreover, DESIRE shows higher robustness than other comparing methods in terms of modeling multi-modalities (Fig. 1 - Row 5) and interaction with other agents (Fig. 1 - Row 6, 7).

2.2. Qualitative Examples for SDD Results

Compared to KITTI Dataset, SDD is a much larger dataset that contains a larger number agents and more complex interactions between agents. In order to highlight the differences and show the improvement of DESIRE over other methods, we present qualitative examples by grouping them into two categories: 1) Fig. 2 - Left: DESIREs present accurate long-term predictions in that they accumulate potential future rewards via IOC framework. On the other hand, RNN Encoder-Decoders do not tend to incorporate scene context well in advance, producing much reactive predictions (*e.g*., predictions are often made over a region that is not crossable or not frequently visited). 2) Fig. 2 - Right: Compared to DESIRE-S, DESIRE-SI accounts for potential interaction between agents while making predictions. For example, Row 1, 5 in Fig. 2 - (Right) show DESIRE-SI's capability of avoiding other agents, and Row 2, 3, 6, 7, 8 in Fig. 2 - (Right) show DESIRE's characteristic of being able to follow other agents in consideration of scene context together.

We provide more qualitative examples in Fig. 3 to further show the reliability of DESIRE under various situations. DESIRE shows superior prediction performance over other methods by reasoning about multiple cues (*i.e*., past motions, scene context and interactions) that affect the agents' behavior in the future. For example, Row 1, 2, 3, 4 in the left of Fig. 3 show that DESIRE-SI predicts more accruately than other methods, by anticipating the predictive behavior of a group of other agents present in the scene. To be more specific, Row 2, 3 in the right of Fig. 3 show that DESIRE-SI produces more accurate prediction by avoiding a potential collision to other agents. In addition, it is also seen from the last rows in both Left and Right of Fig. 3 that DESIRE-SI successfully models to predict behaviors of multiple agents, *i.e*., two close agents present similar motions that follow each other.

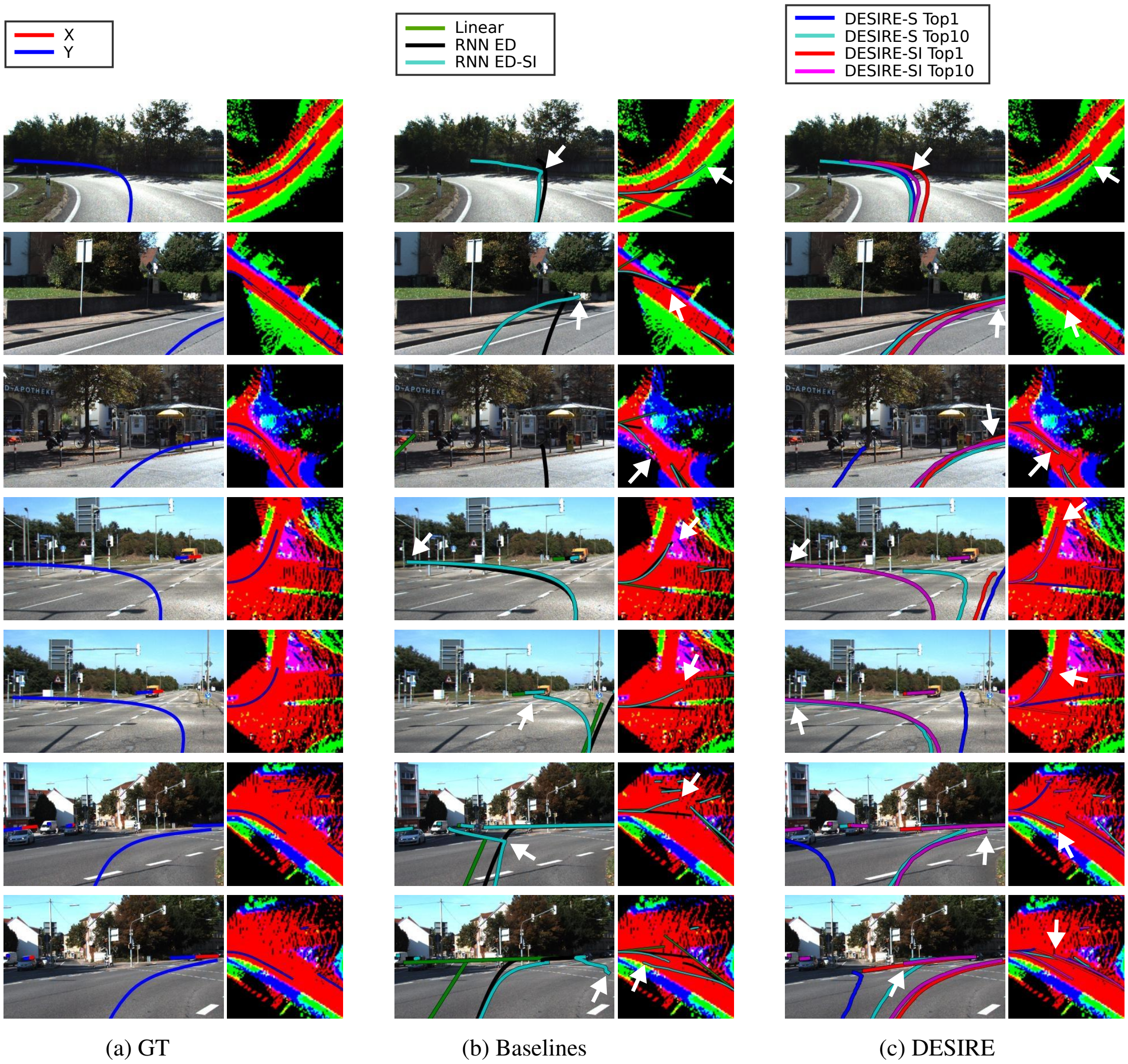
2.3. Failure Cases

DESIRE produces multiple prediction hypotheses to address the uncertainty inherent in future prediction during the sample generation process. Admittedly, however, there are some cases in which DESIRE does not produce a prediction close to the ground-truth. For example, DESIRE may not cover all plausible paths where multiple choices are valid (*e.g*., DESIRE chooses an adjacent exit when approaching to a large round-about (*e.g*., Row 3 in Fig. 5 - Left)). In addition, DESIRE often fails to generate the ground truth prediction for an agent when the agent is making a sudden route change (*e.g*., an agent suddenly turns around or deviates the course (*e.g*., Row 1, 2 in Fig. 5 - Right)). We provide some qualitative examples where DESIRE fails to produce accurate predictions for both KITTI and SDD, in Fig. 4 and Fig. 5, respectively.

References

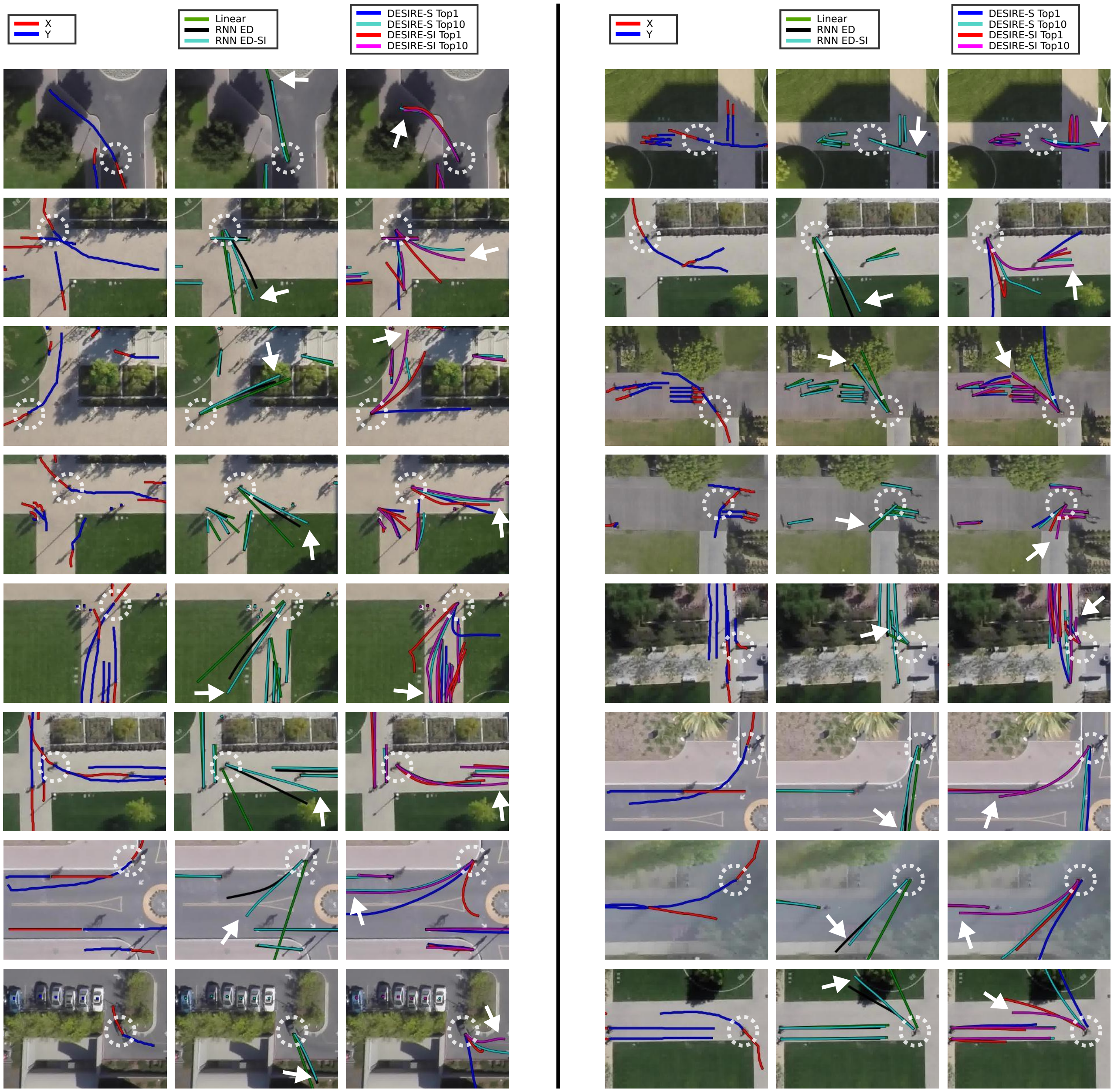
[1] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 1

[2] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1



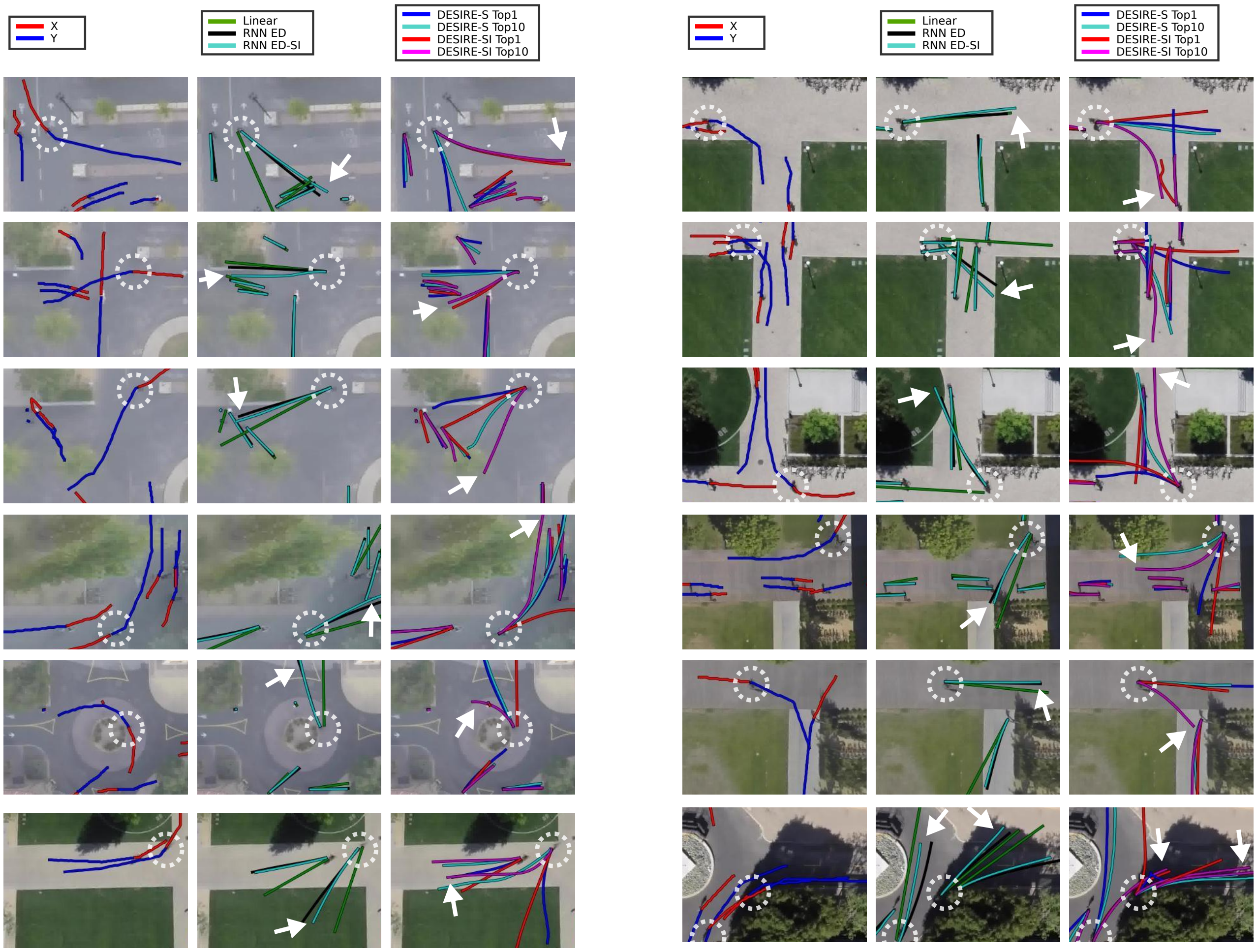
(a) GT (b) Baselines (c) DESIRE

Figure 1. Qualitative examples for KITTI results. (Row 1, 2, 3, 4) The predictions produced by RNN Encoder-Decoders (middle) are too reactive (*i.e*., they often make sudden turns when the vehicle almost hits non-drivable region). On the other hand, the predictions produced by DESIRE are smooth and more accurate to the ground-truth as it considers long-term future rewards. (Row 5) There are three routes available, and DESIRE produces a correct path prediction, whereas RNN Encoder-Decoder chooses an incorrect path which is far from the ground-truth. (Row 6, 7) DESIRE-SI produces more accurate predictions than other baselines in the presence of other agents. DESIRE-SI exploits the interaction between other agents far better than other baselines through the Scene Context Fusion.



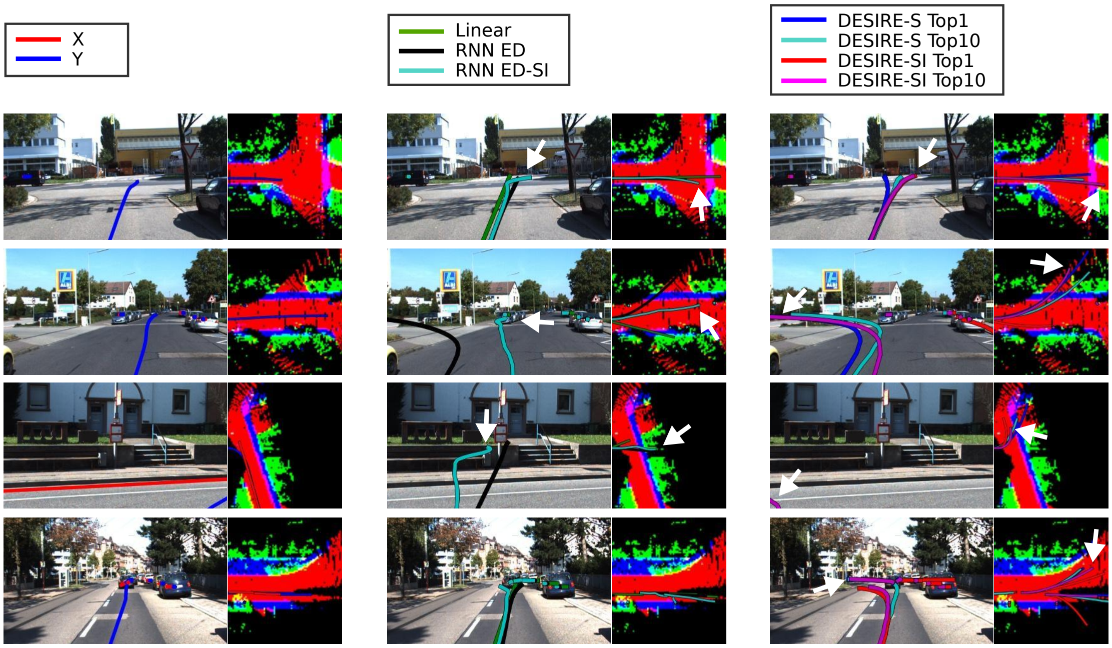
(a) GT (b) Baselines (c) DESIRE (a) GT (b) Baselines (c) DESIRE

Figure 2. Qualitative examples for SDD results. (Left) Comparisons between RNN Encoder-Decoders and DESIREs. Note that DESIREs result in predictions much closer to the actual ground-truth as they reflect scene context much better than RNN Encoder-Decoders via IOC framework. (Right) Comparisons between DESIRE-SI and DESIRE-S. DESIRE-SI incorporates interaction between agents as well as the static scene context, producing more accurate prediction results than DESIRE-S.



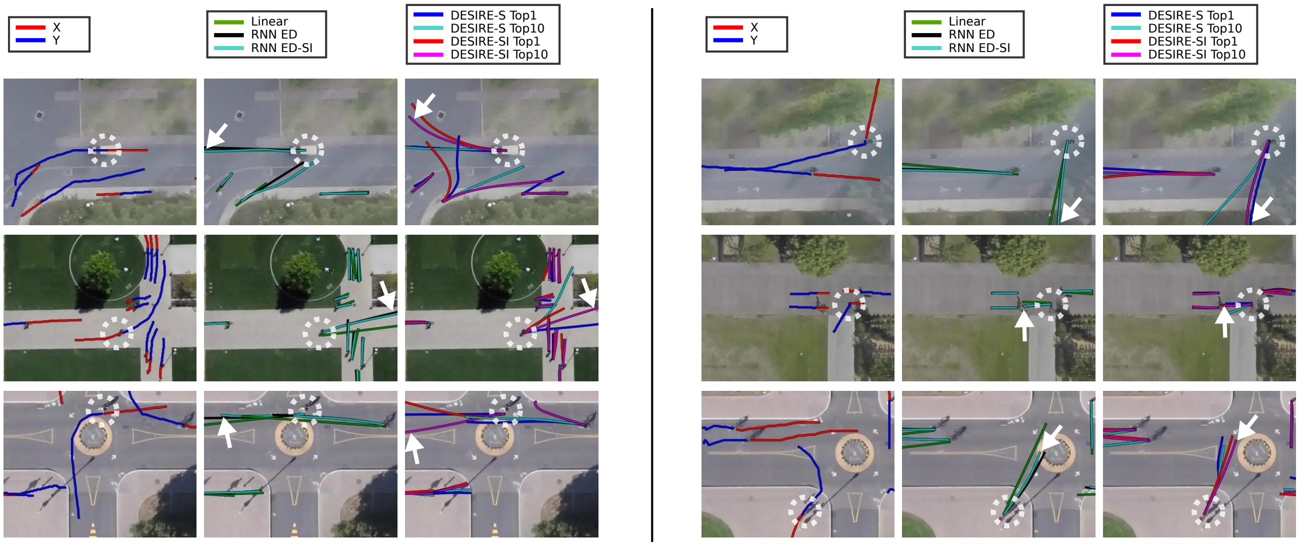
(a) GT (b) Baselines (c) DESIRE (a) GT (b) Baselines (c) DESIRE

Figure 3. Additional qualitative examples for SDD results. DESIRE-SI enables more accurate predictions compared to RNN Encoder-Decoders or DESIRE-S, as it produces the future prediction by jointly reasoning from various cues that affect the behavior of multiple agents, such as the past dynamics, the static scene context and the interaction between agents.



(a) GT (b) Baselines (c) DESIRE

Figure 4. Selected failure cases from KITTI Dataset. Due to multi-modalities inherent in the future prediction, DESIRE may not provide the GT prediction when there are multiple plausible paths. (Row 1) For a car approaching toward an intersection, DESIRE predicts to keep forward. (Row 2) DESIRE chooses to make a left turn to enter a parking lot. (Row 3) DESIRE chooses to make a left turn whereas the actual ground-truth makes a right turn. (Row 4) There are two possible paths for a vehicle, and DESIRE chooses to take a left lane.



(a) GT (b) Baselines (c) DESIRE (a) GT (b) Baselines (c) DESIRE

Figure 5. Selected failure cases from SDD. (Left) DESIRE may not always produce a prediction close to the ground-truth when multiple plausible paths are valid (*i.e.*, multi-modalities of future prediction). (Right) DESIRE may not produce accurate prediction under abrupt motion changes.