



Carnegie  
Mellon  
University



# EAGLE : Efficient Adaptive Geometry-based Learning in Cross-view Understanding

Thanh-Dat Truong, Utsav Prabhu, Dongyi Wang

Bhiksha Raj, Susan Gauch, Jeyamkondan Subbiah, Khoa Luu

CVIU Lab, University of Arkansas

Google DeepMind

University of Arkansas

Carnegie Mellon University

MBZUAI

<https://uark-cviu.github.io/projects/EAGLE/>



Arkansas  
BIOSCIENCES  
INSTITUTE 



# Cross-view Adaptation

find cars, persons, trees  
from the car view

CLIP Text  
Encoder

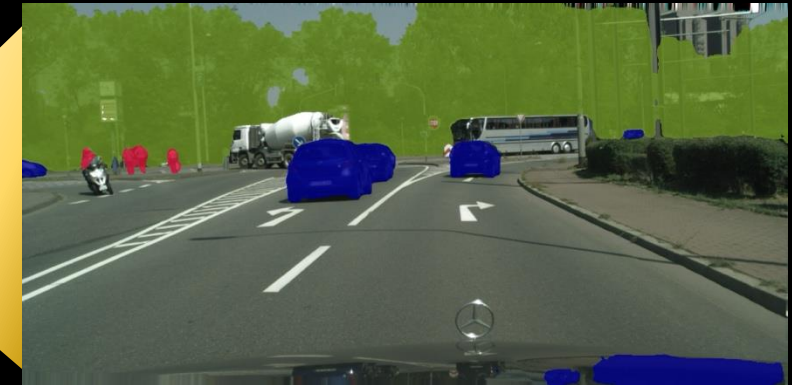
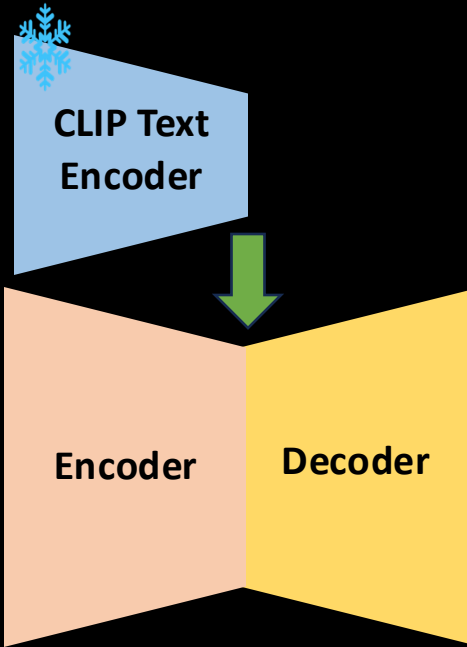
Encoder

Decoder

car

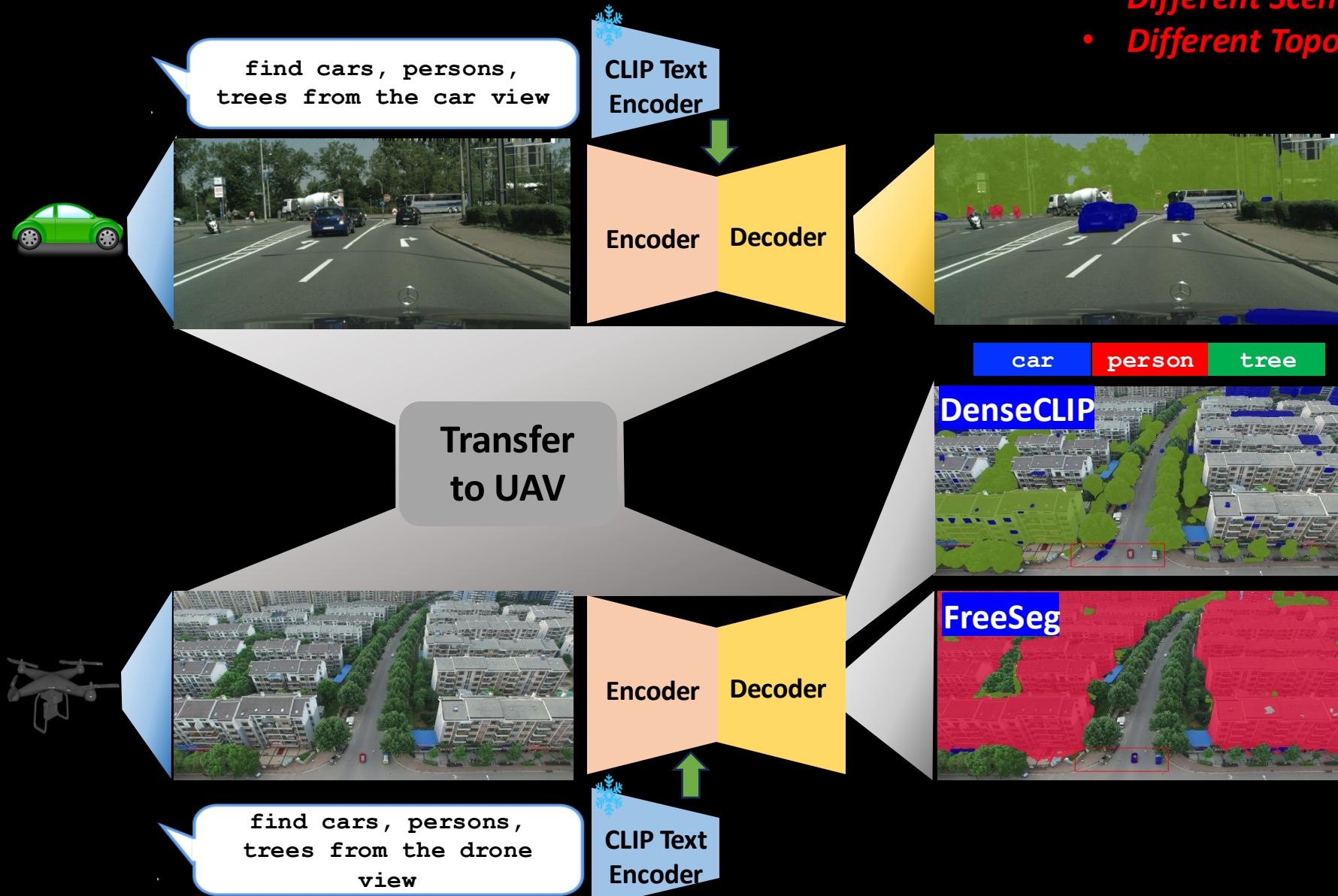
person

tree



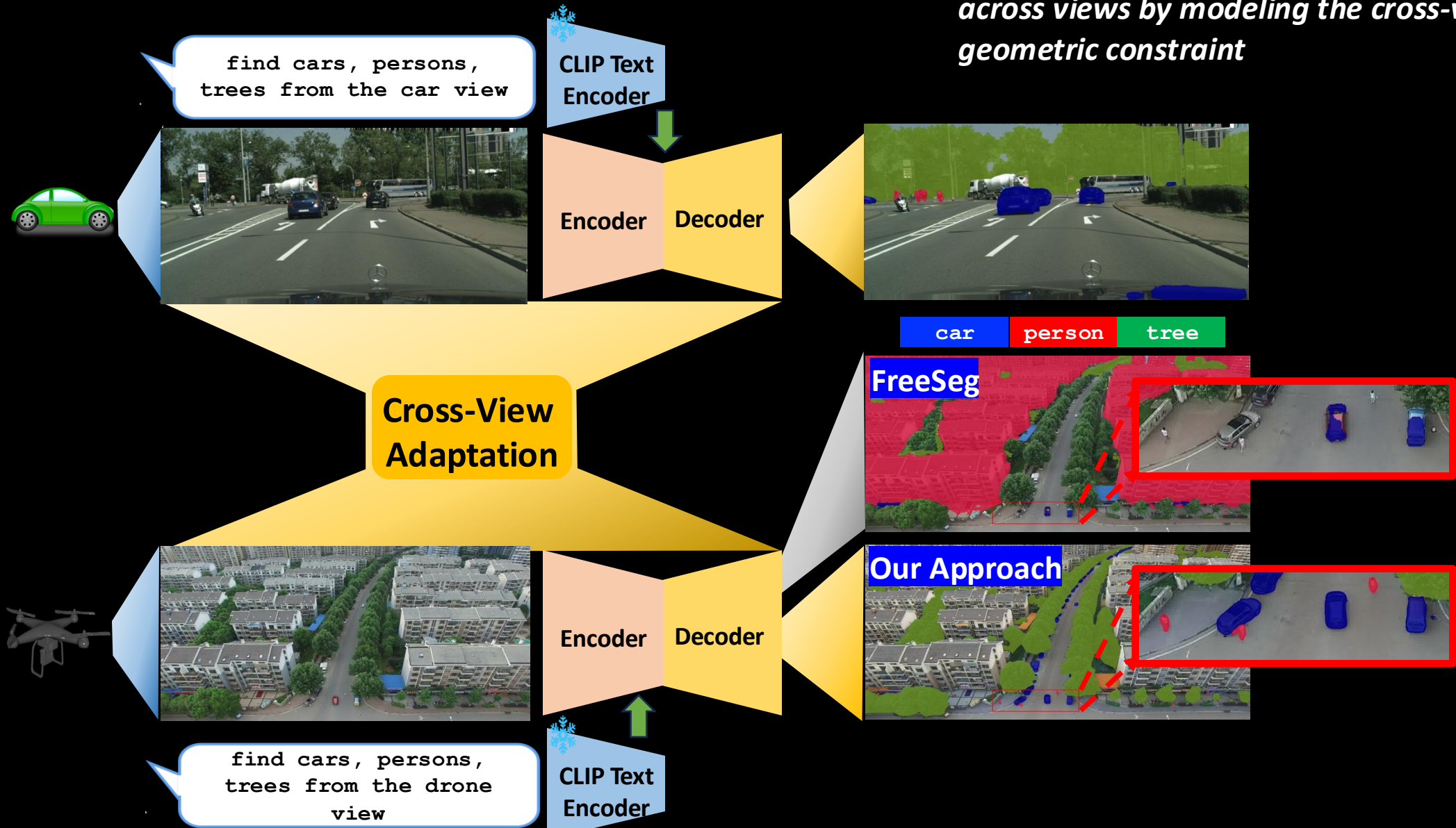
# Cross-view Adaptation

- Worse Performance due to*
- Different Camera Position*
  - Different Scene Structures*
  - Different Topological Layout*



# Cross-view Adaptation

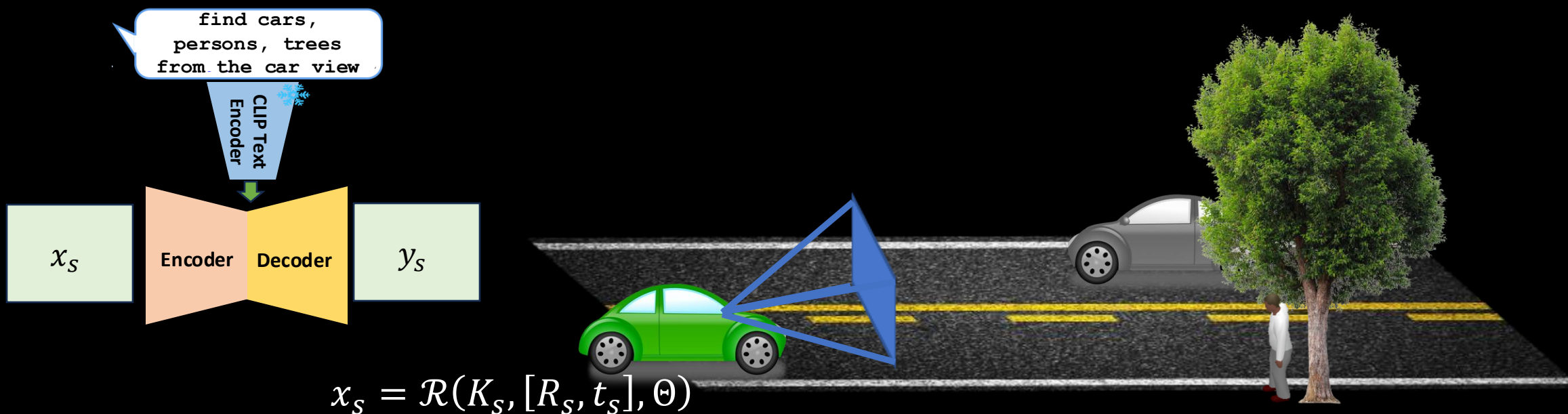
✓ *Our Cross-view Adaptation Approach improves the performance of segmentation across views by modeling the cross-view geometric constraint*



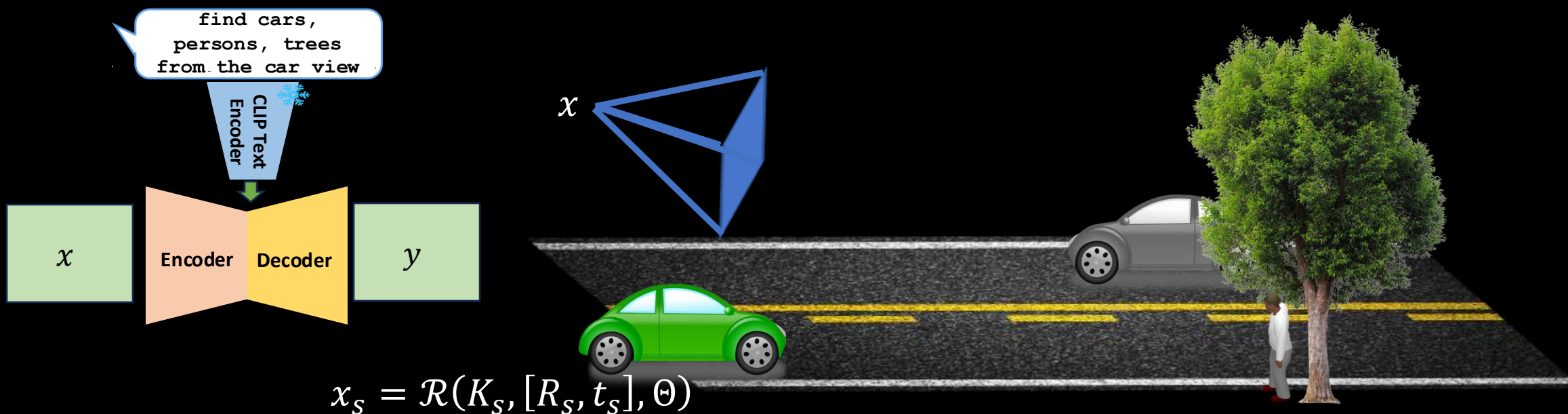
# Contributions

- ✓ Introduce a novel ***Cross-view Adaptation Learning Approach to Semantic Segmentation***
  - ✓ Present a ***new cross-view geometric constraints*** between image and segmentation spaces
  - ✓ Propose a new ***Geodesic Flow-based metric*** to measure the ***cross-view structural changes***
  - ✓ Introduce ***a novel view-condition prompting mechanism*** to cross-view adaptation learning
- ✓ Achieve the ***State-of-the-Art performance*** compared to prior adaptation approaches and ***improve robustness of semantic segmentation models across views***

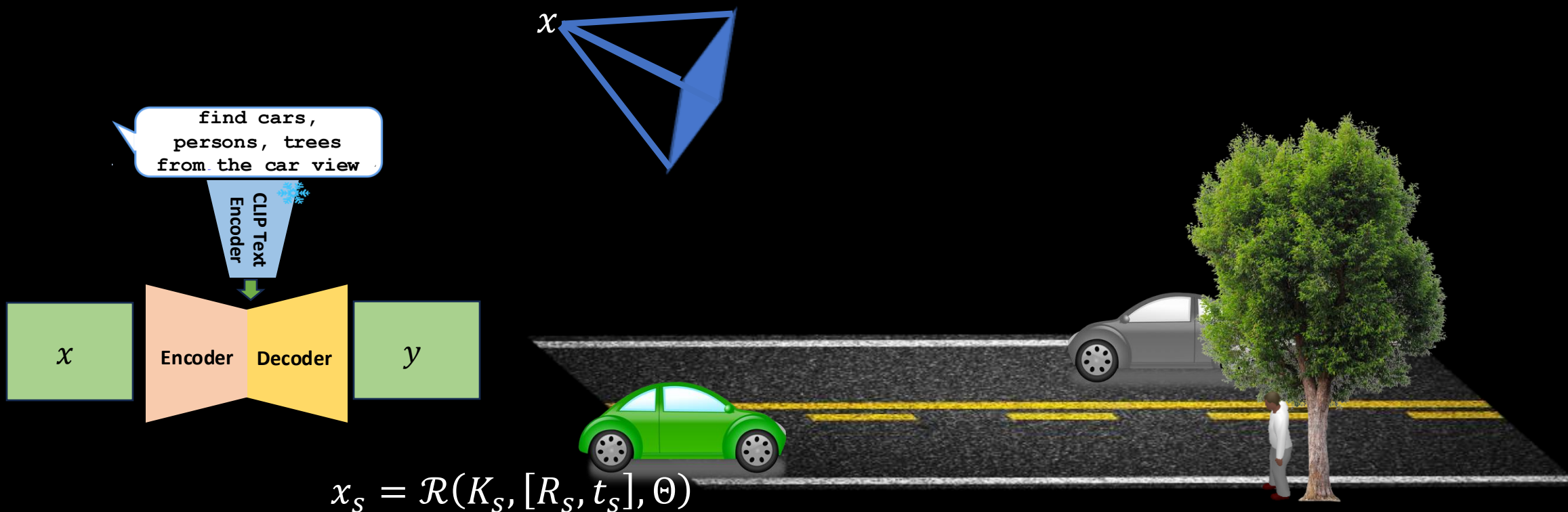
# Cross-view Geometric Constraint



# Cross-view Geometric Constraint

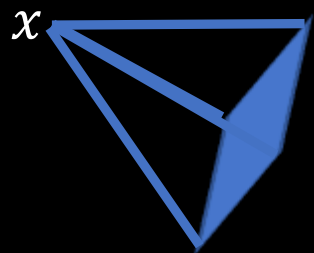


# Cross-view Geometric Constraint



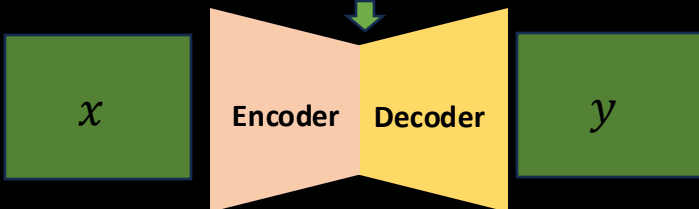


# Cross-view Geometric Constraint



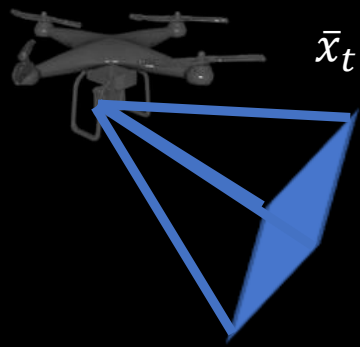
find cars,  
persons, trees  
from the car view

CLIP Text  
Encoder



$$x_s = \mathcal{R}(K_s, [R_s, t_s], \Theta)$$

# Cross-view Geometric Constraint



$$\bar{x}_t = \mathcal{R}(K_t, [R_t, t_t], \Theta)$$

find cars,  
persons, trees  
from the car view

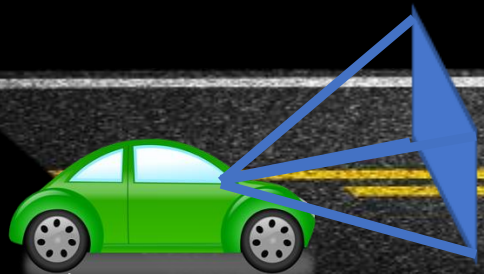
CLIP Text  
Encoder

$\bar{x}_t$

Encoder

Decoder

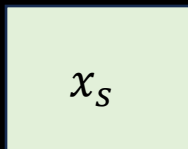
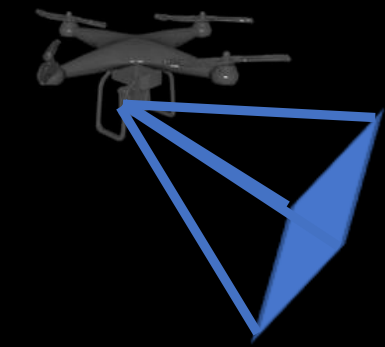
$\bar{y}_t$



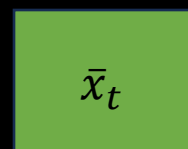
$$x_s = \mathcal{R}(K_s, [R_s, t_s], \Theta)$$



# Cross-view Geometric Constraint



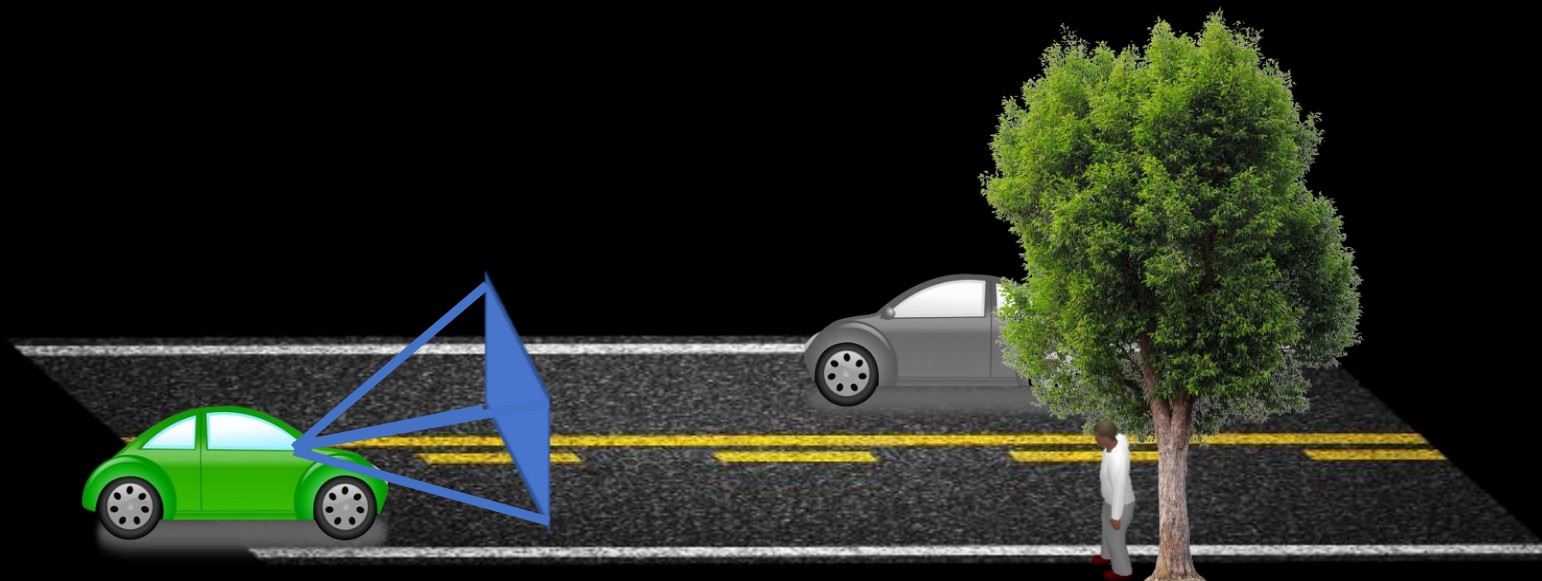
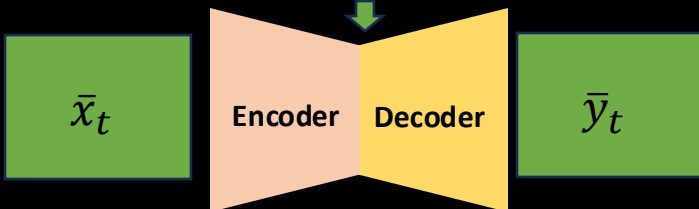
$$x_s = \mathcal{R}(K_s, [R_s, t_s], \Theta)$$



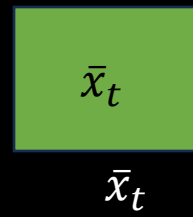
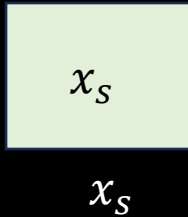
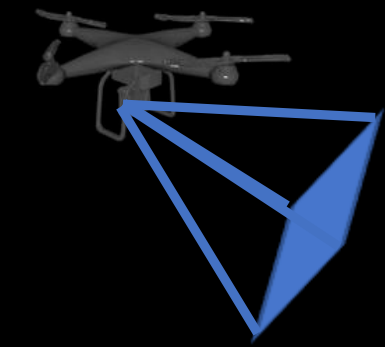
$$\bar{x}_t = \mathcal{R}(K_t, [R_t, t_t], \Theta)$$

find cars,  
persons, trees  
from the car view

CLIP Text  
Encoder



# Cross-view Geometric Constraint

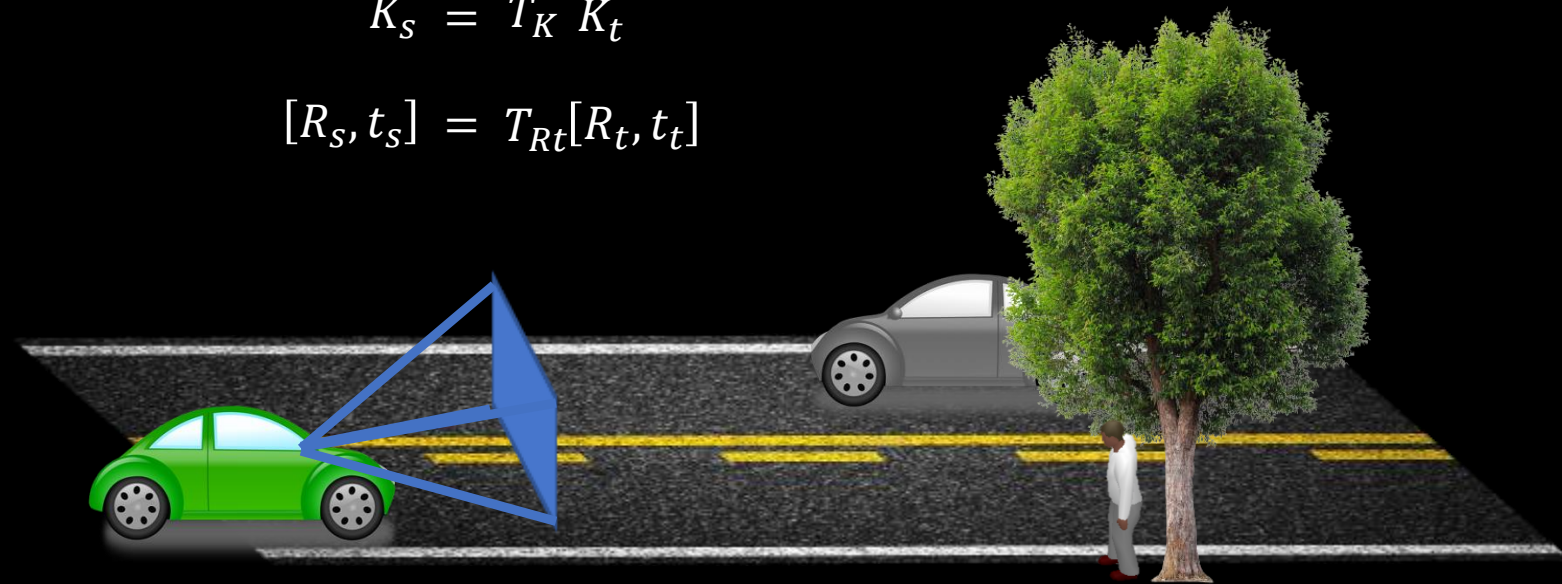
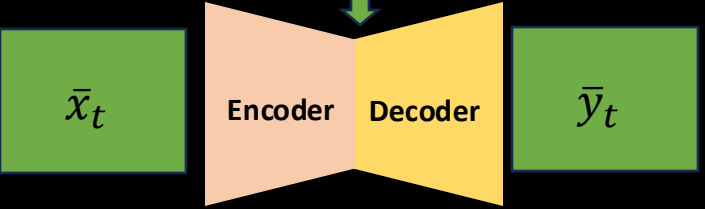


$$K_s = T_K K_t$$

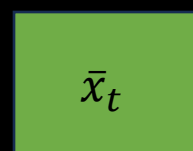
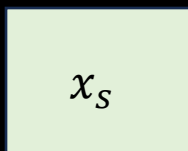
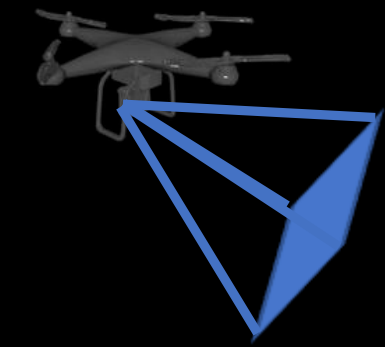
$$[R_s, t_s] = T_{Rt}[R_t, t_t]$$

find cars,  
persons, trees  
from the car view

CLIP Text  
Encoder



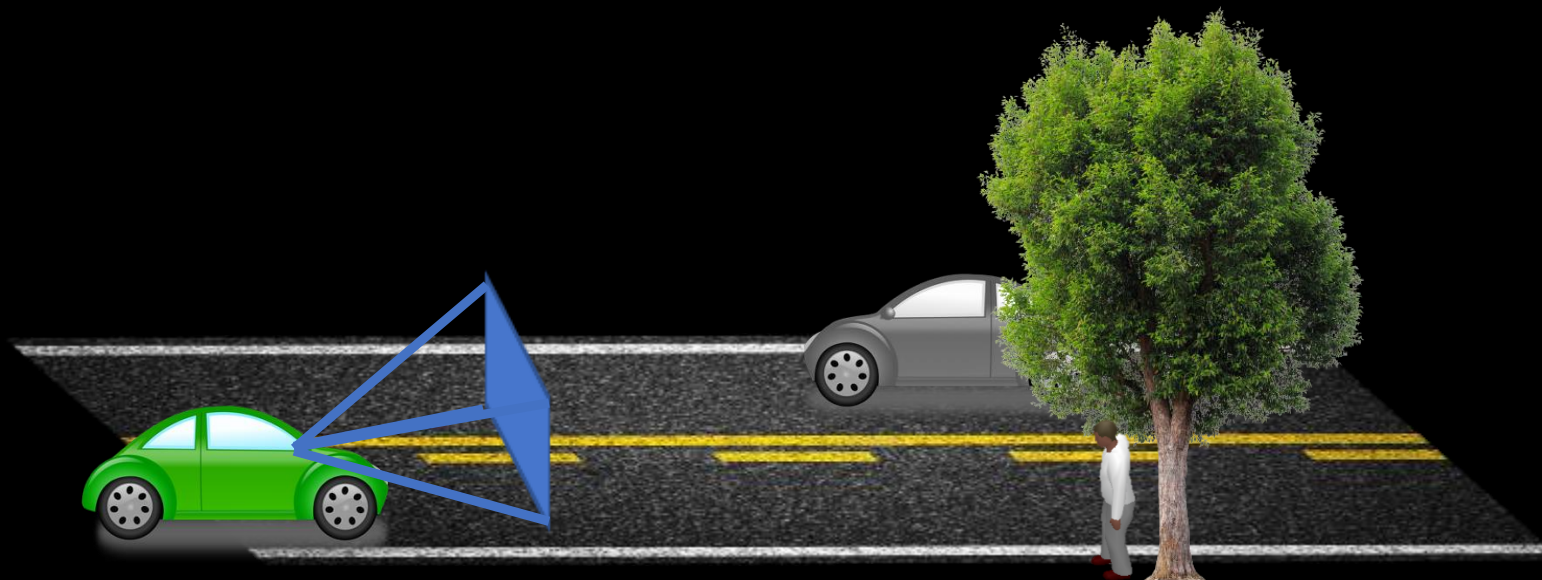
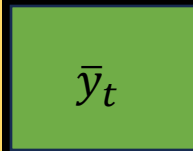
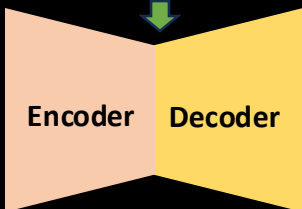
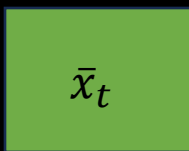
# Cross-view Geometric Constraint



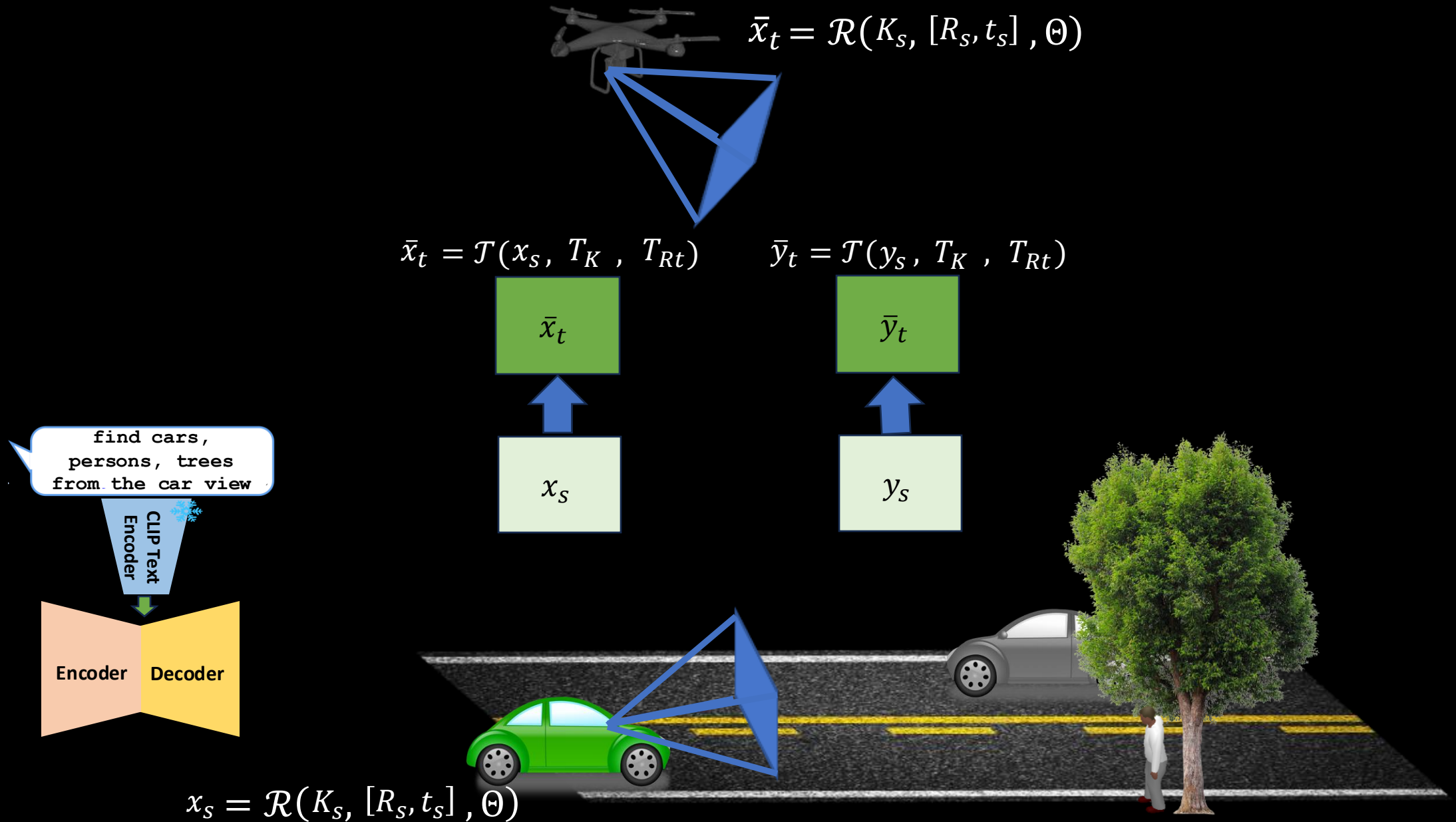
$$\bar{x}_t = \mathcal{T}(x_s, T_K, T_{Rt})$$

find cars,  
persons, trees  
from the car view

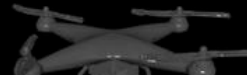
CLIP Text  
Encoder



# Cross-view Geometric Constraint



# Cross-view Geometric Constraint


$$\bar{x}_t = \mathcal{R}(K_t, [R_t, t_t], \Theta)$$

*Measure Cross-view  
Structural Changes in the  
Segmentation Space*

$$\mathcal{D}_x \left( \begin{array}{|c|} \hline x_s \\ \hline \end{array}, \begin{array}{|c|} \hline \bar{x}_t \\ \hline \end{array} \right) = \alpha \mathcal{D}_y \left( \begin{array}{|c|} \hline y_s \\ \hline \end{array}, \begin{array}{|c|} \hline \bar{y}_t \\ \hline \end{array} \right)$$

*Measure Cross-view  
Structural Changes in  
the Image Space*

find cars,  
persons, trees  
from the car view

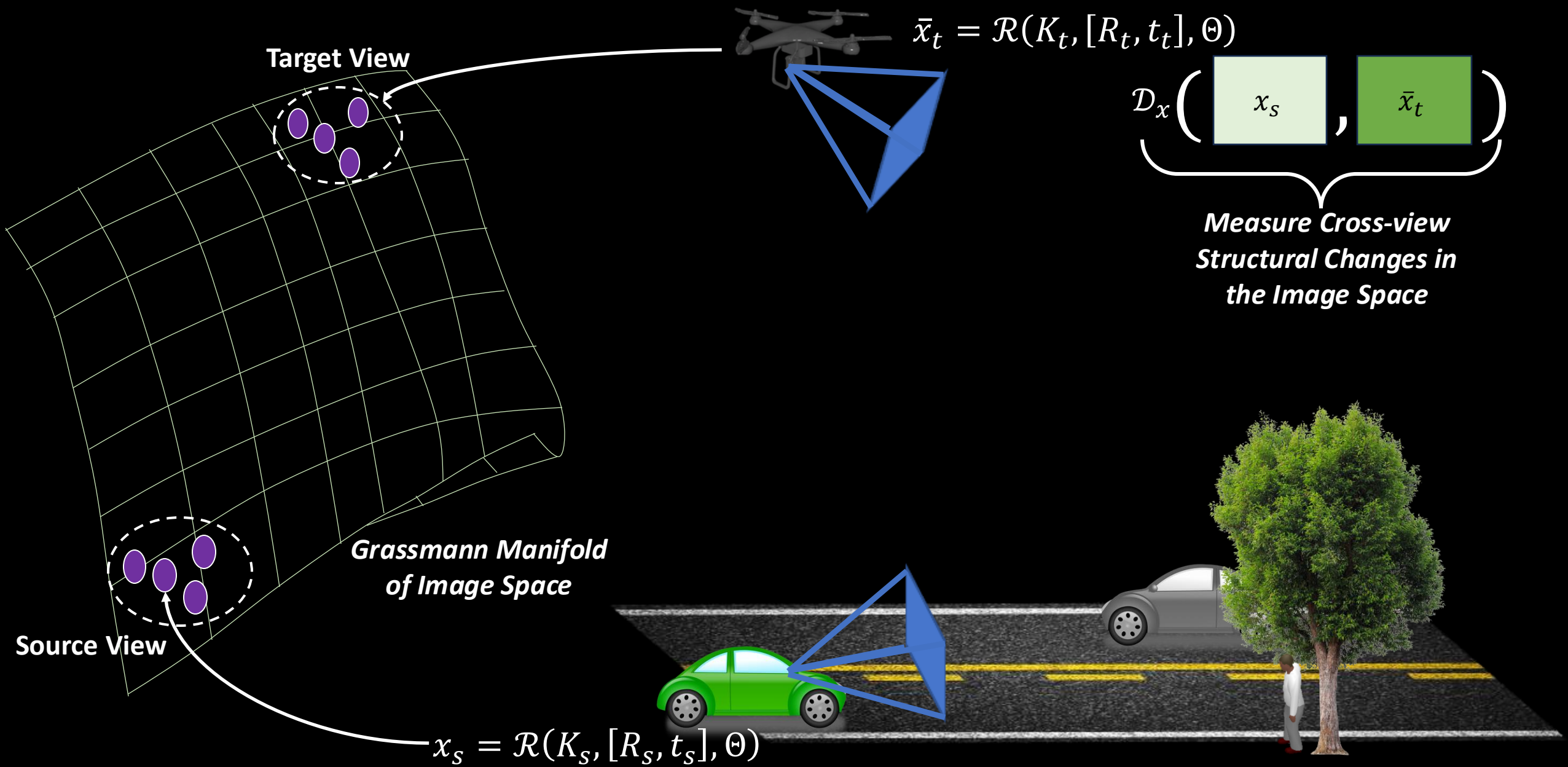
CLIP Text  
Encoder

Encoder

Decoder

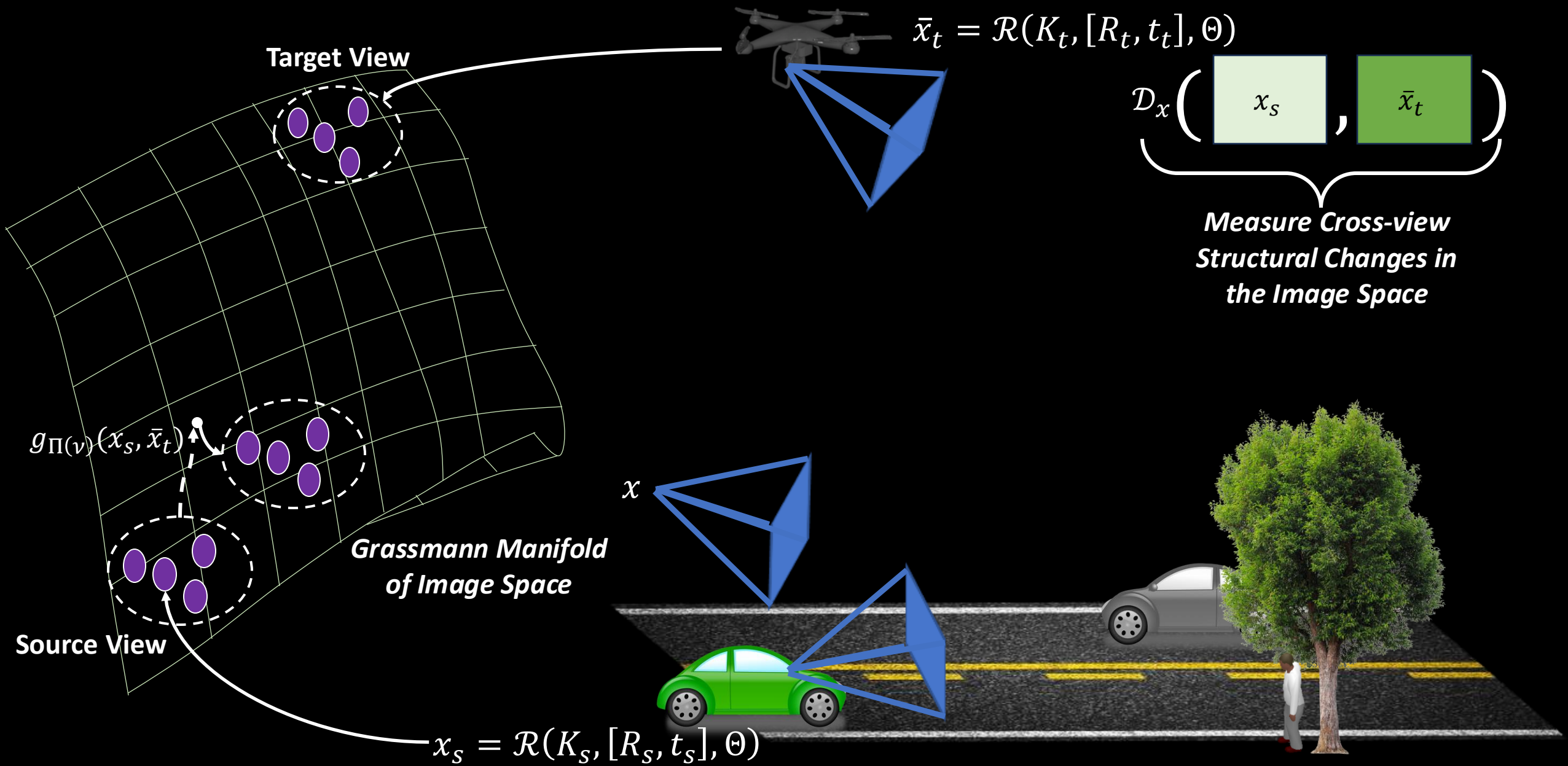

$$x_s = \mathcal{R}(K_s, [R_s, t_s], \Theta)$$

# Cross-view Geometric Constraint

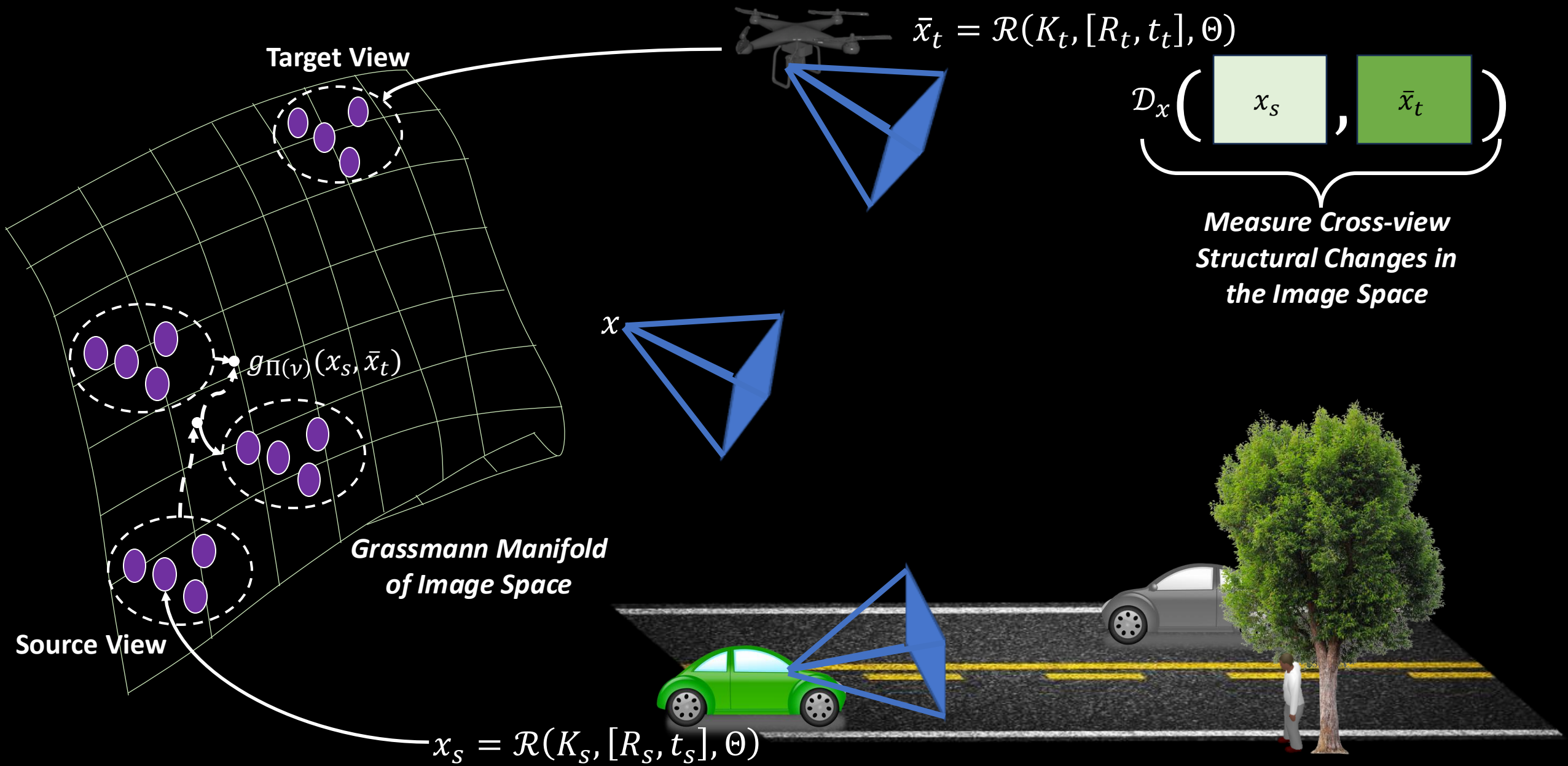




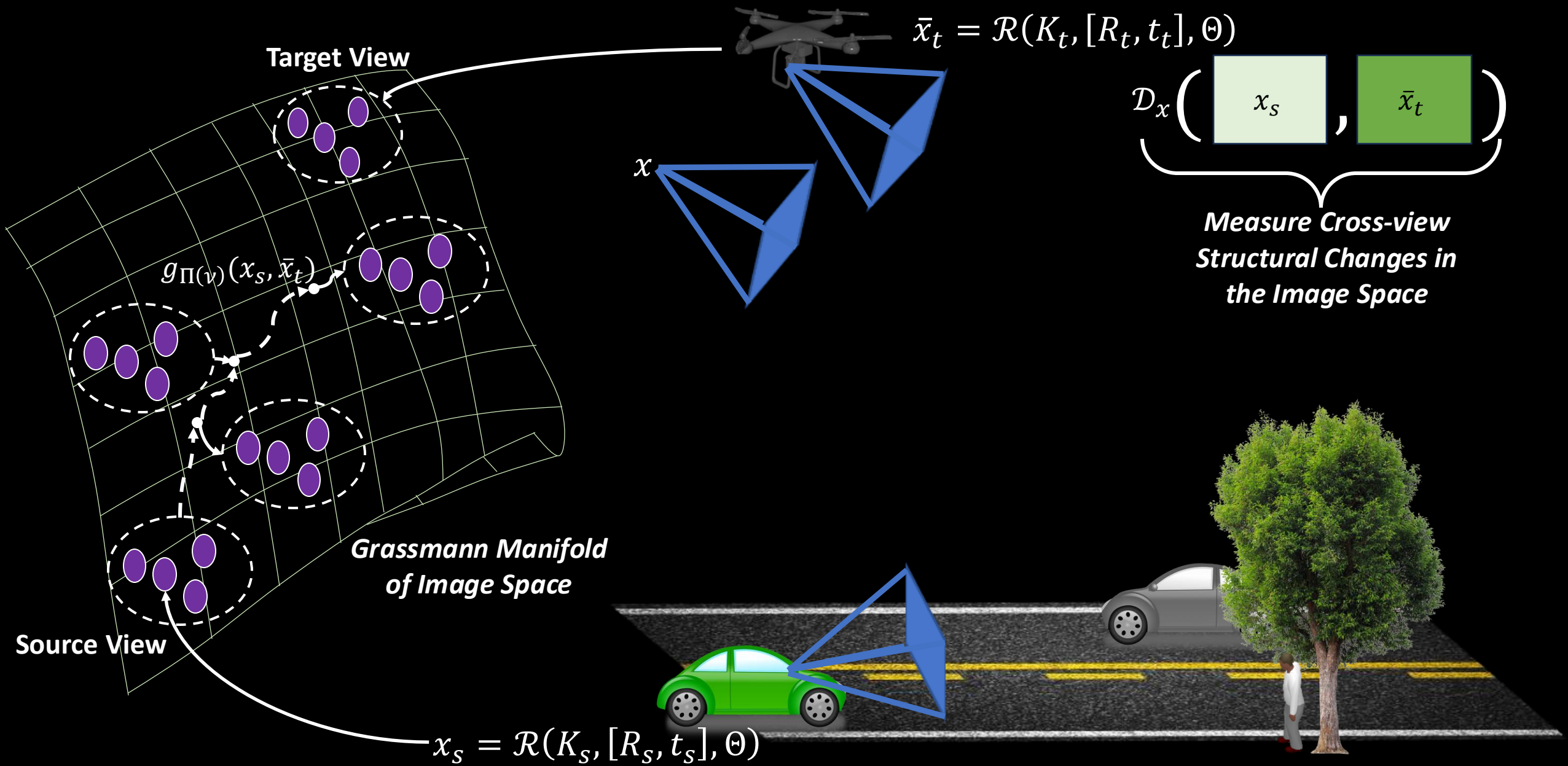
# Cross-view Geometric Constraint



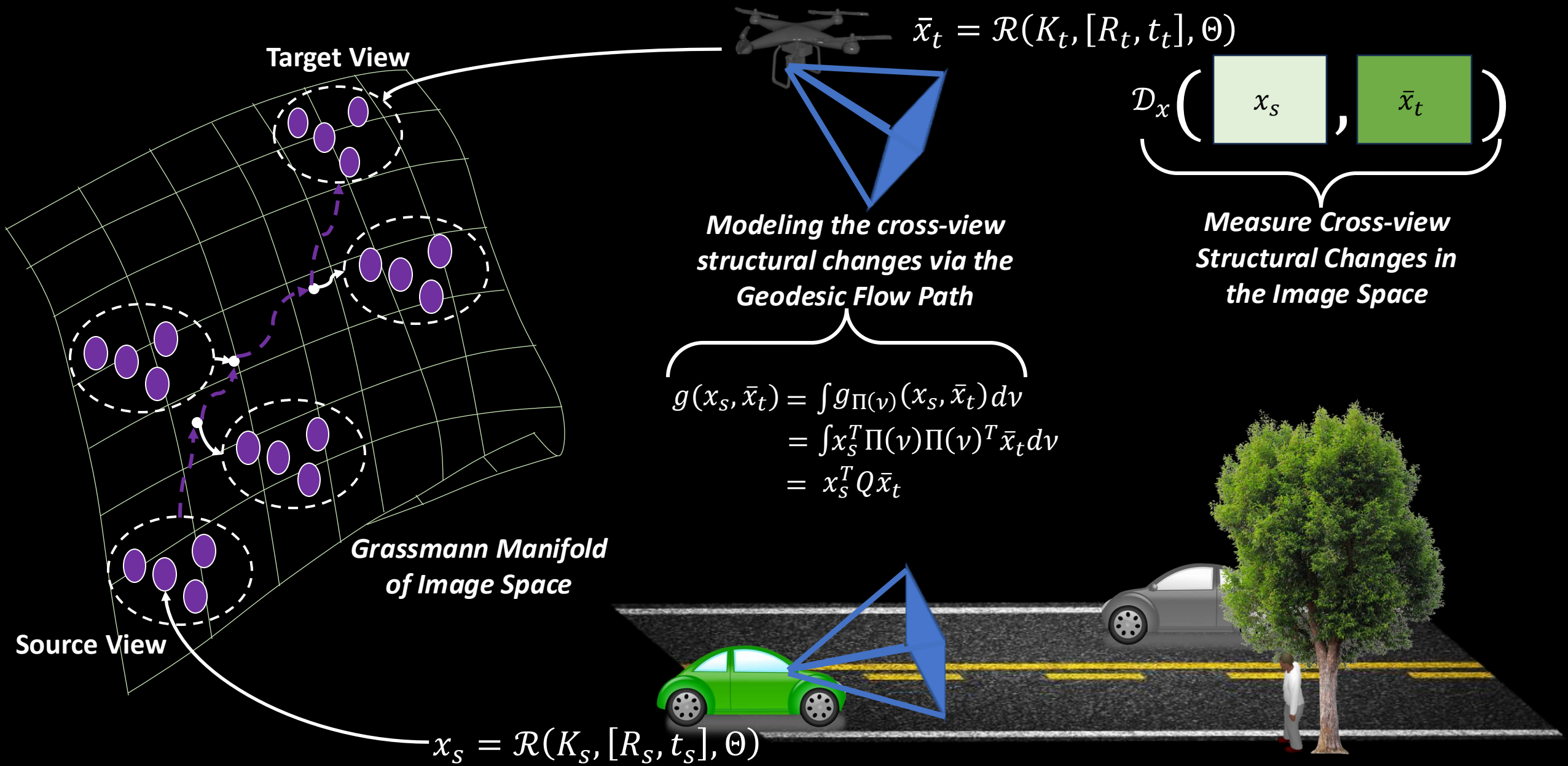
# Cross-view Geometric Constraint



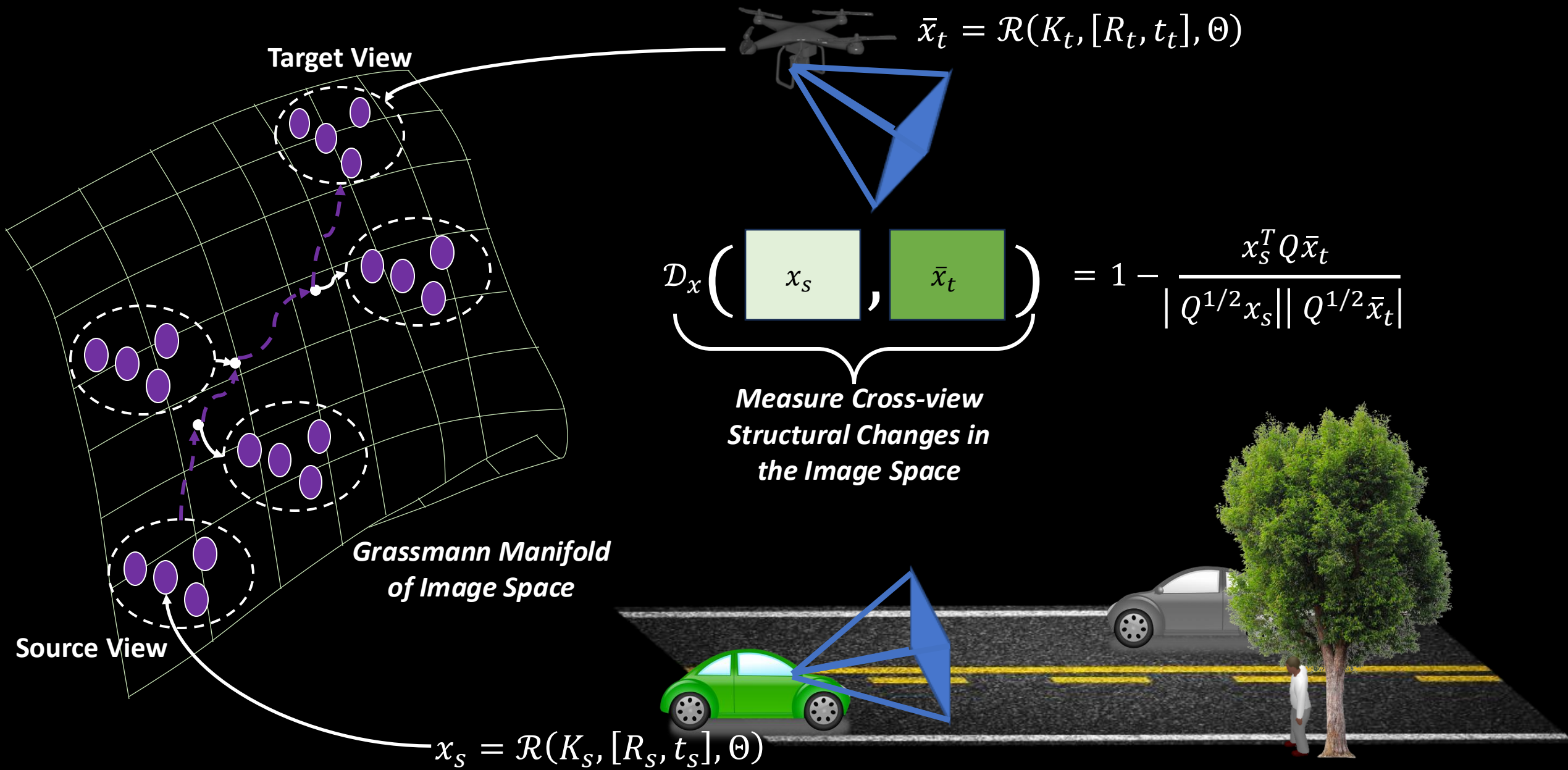
# Cross-view Geometric Constraint



# Cross-view Geometric Constraint



# Cross-view Geometric Constraint



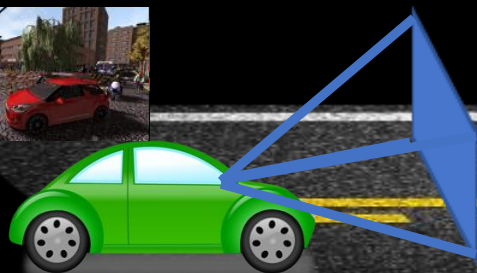
# Cross-view Geometric Constraint

*However, in practice, images of source and target views are collected independently (unpaired data)*

Images collected from the target view



Images collected from the source view



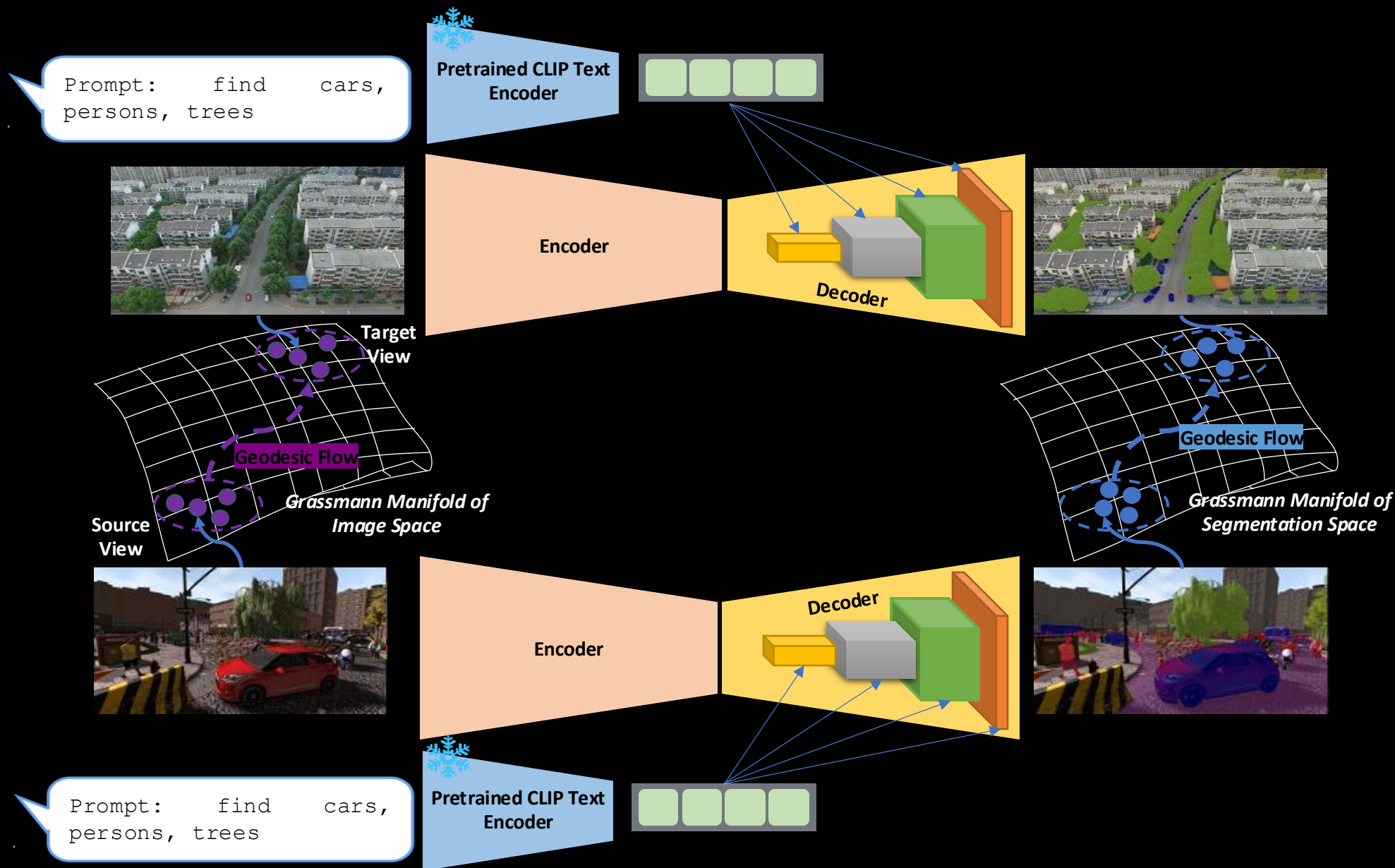
# Cross-view Geometric Constraint



$$\mathcal{D}_x \left( \left[ \text{Street View} \right], \left[ \text{Aerial View} \right] \right) = \alpha \mathcal{D}_y \left( \left[ \text{Street View} \right], \left[ \text{Aerial View} \right] \right)$$

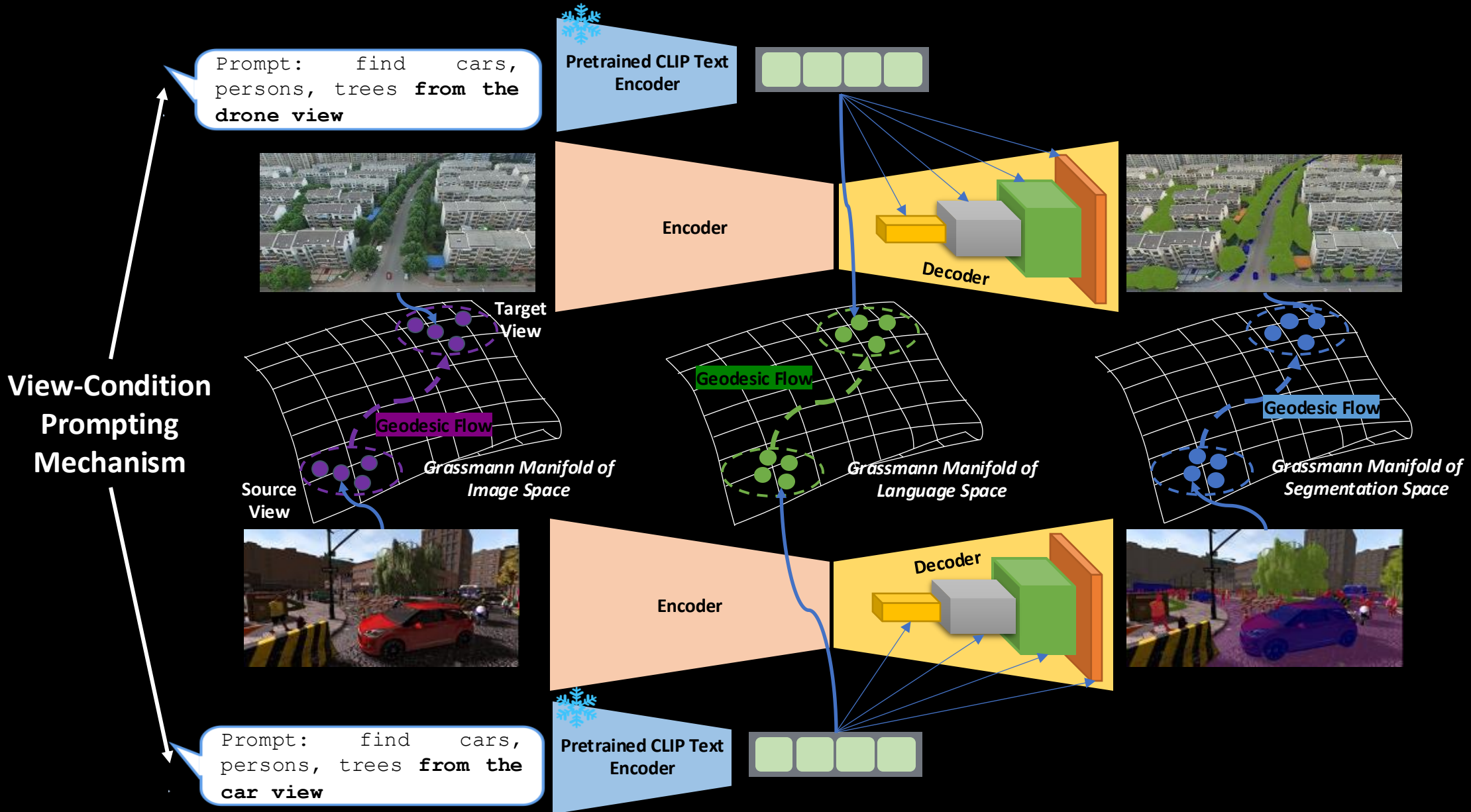
*Modeling the Cross-view Geometric  
Constraint on Unpaired Data*

# Cross-view Geometric Constraint

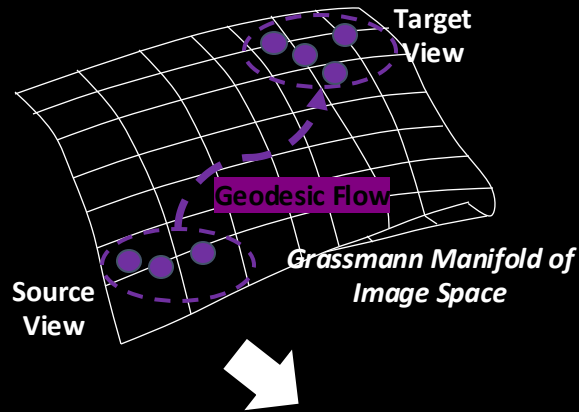




# Cross-view Geometric Constraint

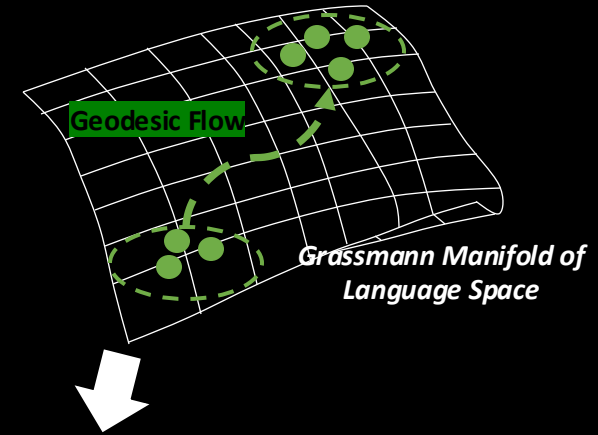


# Cross-view Geometric Constraint



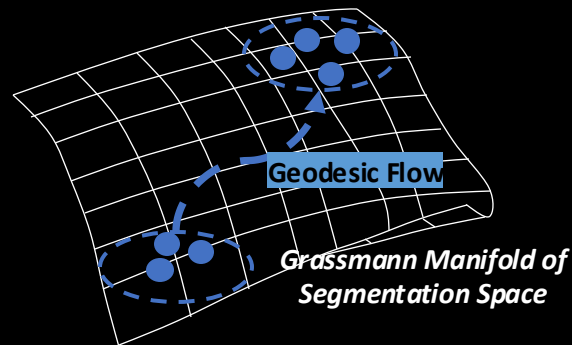
$$|\mathcal{D}_x(x_s, x_t) - \alpha \mathcal{D}_y(y_s, y_t)|$$

**Cross-view Geometric Adaptation Loss  
Guided by the Geodesic Flow in the Image  
Space**



$$|\mathcal{D}_p(f_s^p, f_t^p) - \gamma \mathcal{D}_y(y_s, y_t)|$$

**Cross-view Geometric Adaptation Loss  
Guided by the Geodesic Flow in the  
Language Space**



Video Demo