Towards Automated Testing and Robustification by Semantic Adversarial Data Generation

Rakshith Shetty[1] **Mario Fritz[2]** **Bernt Schiele[1]**

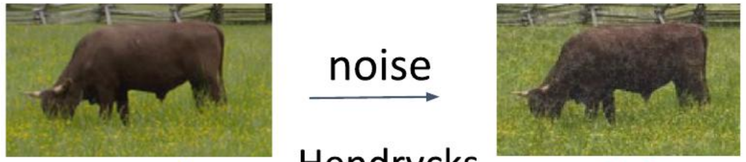
[1]Max Planck Institute for Informatics, Saarland Informatics Campus
[2]CISPA Helmholtz Center for Information Security

IID testing in computer vision is insufficient

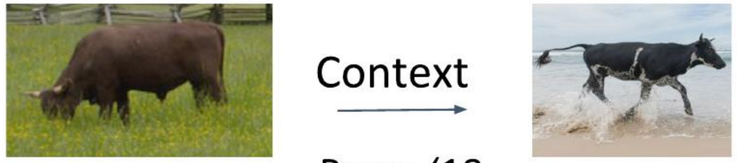
- Everything works well on i.i.d data

Train ✔ i.i.d → Test ✔

- Models break down under different distribution shifts



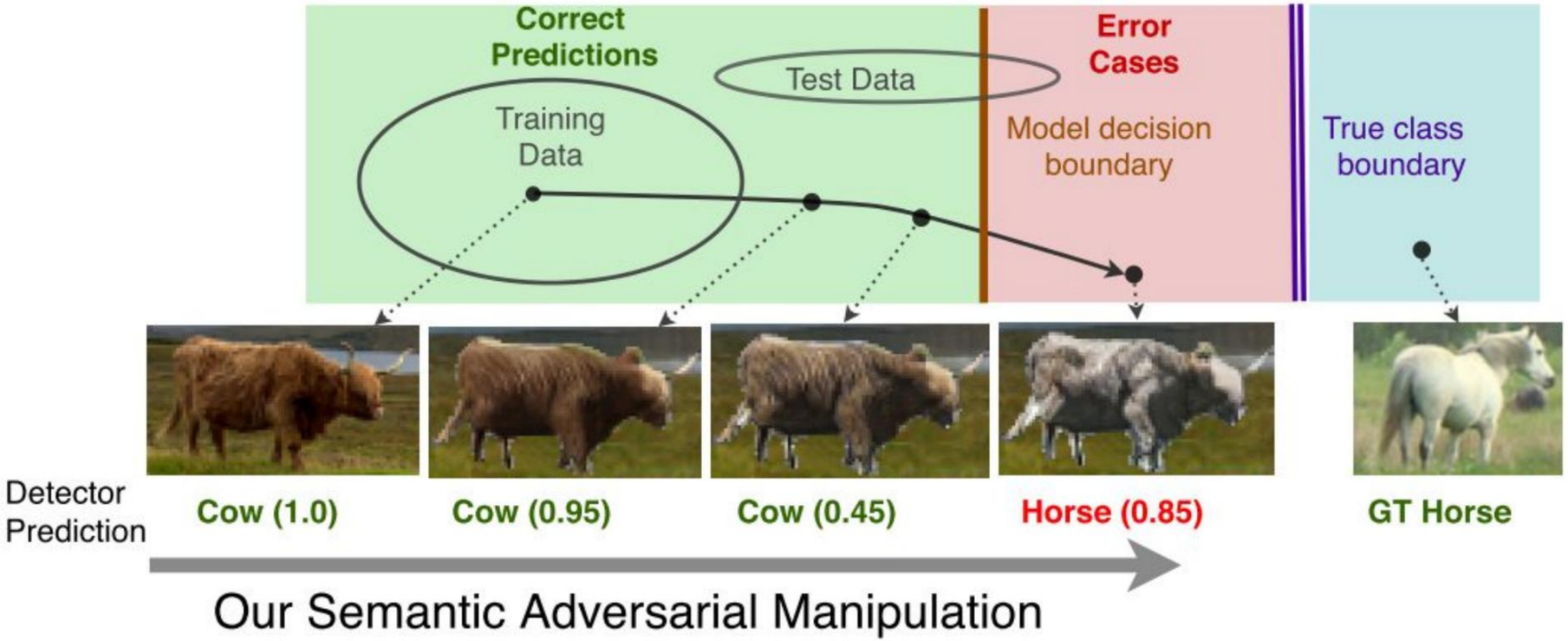
Train ✔ noise → Hendrycks '19 Test ❌



Train ✔ Context → Beery '18 Test ❌

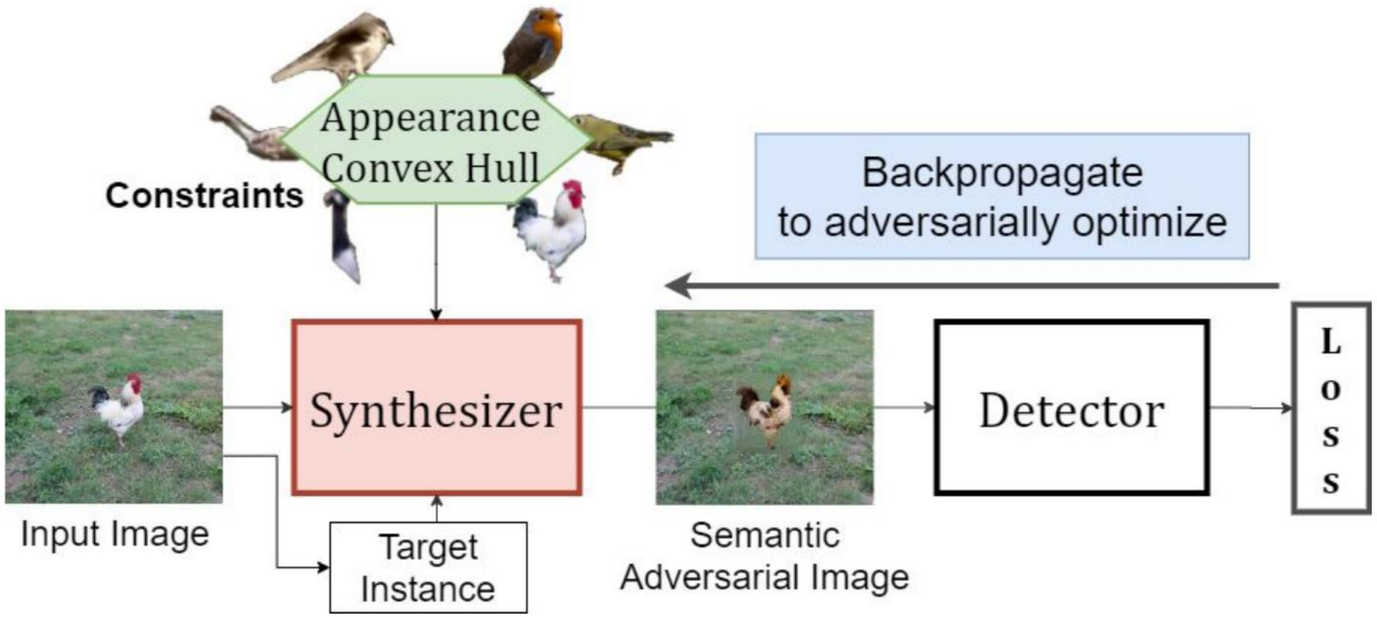
- Evaluating with a finite sample i.i.d test set overestimates performance

Model adaptive testing via semantic adversary



- Core Idea: Use a **generative model** + constrained **adversarial attack** to move in the data space and synthesize targeted novel failure modes

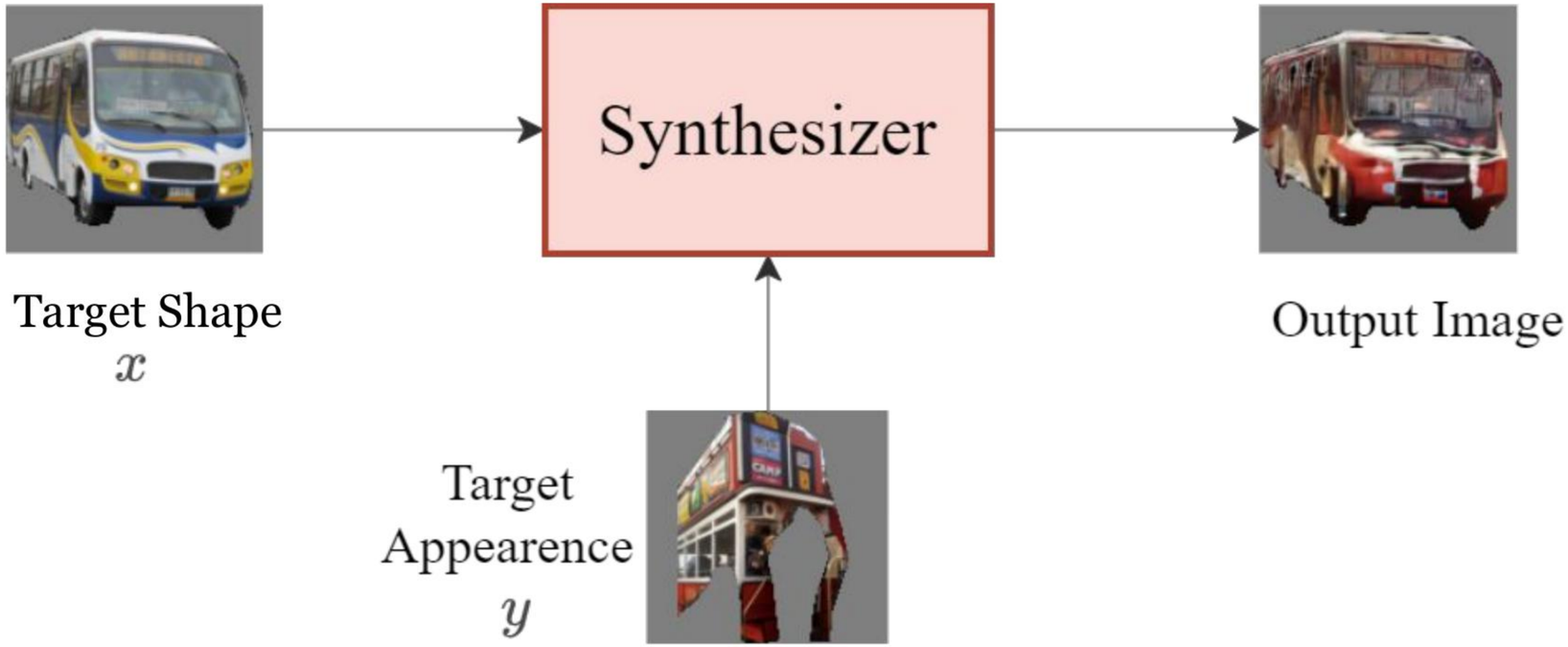
Key components



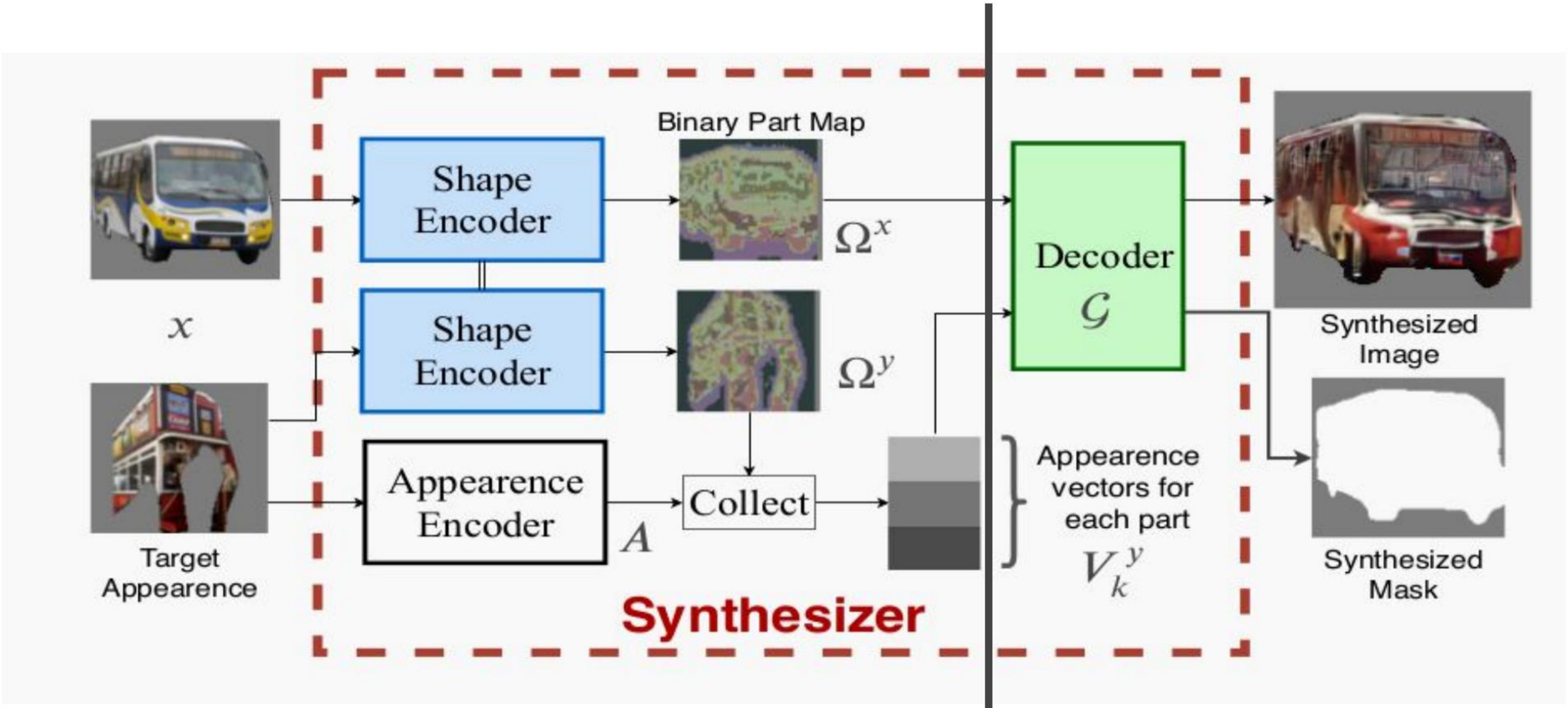
- A **synthesizer** → generates objects with disentangled pose and appearance
- **Adversarial optimization** → guide the synthesis to towards hard cases
- **Appearance constraints** → keep synthesized appearance realistic

Deep dive into the Synthesizer

Synthesizer network

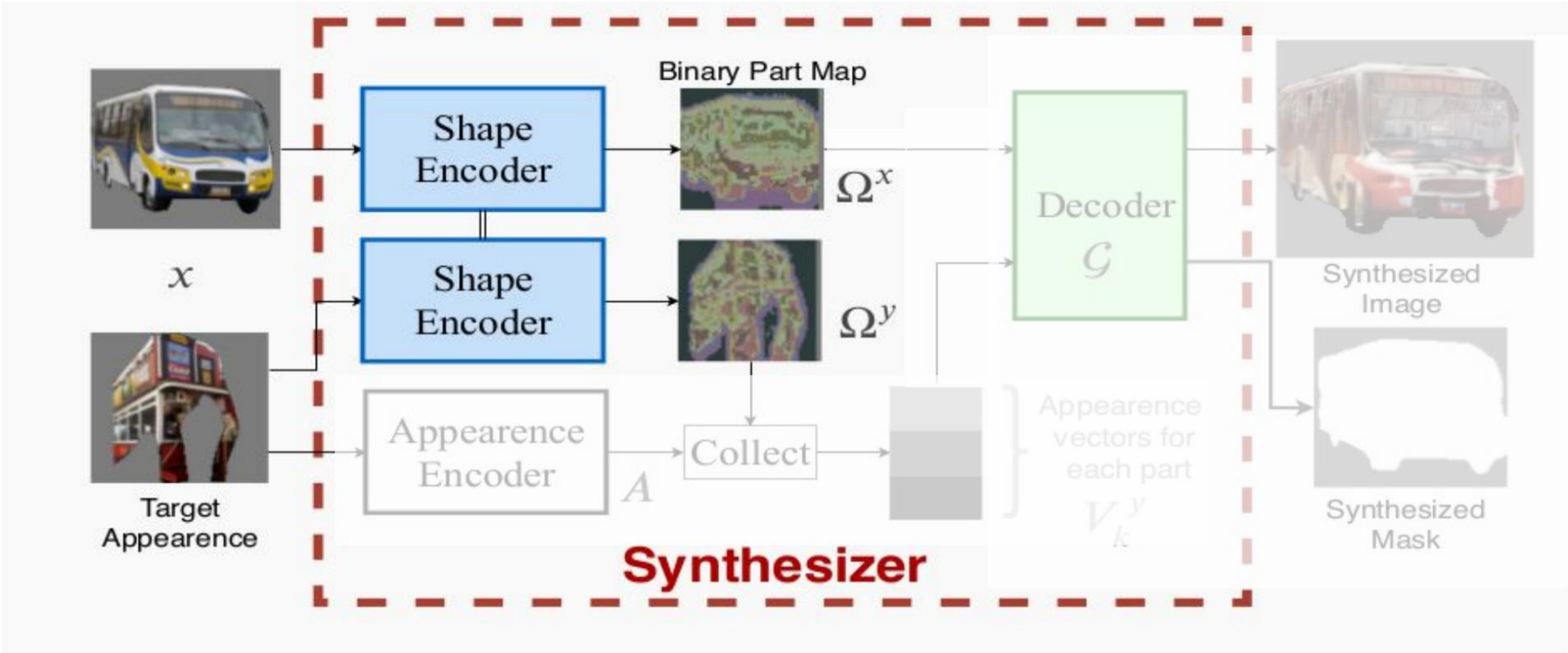


Inside the synthesizer network



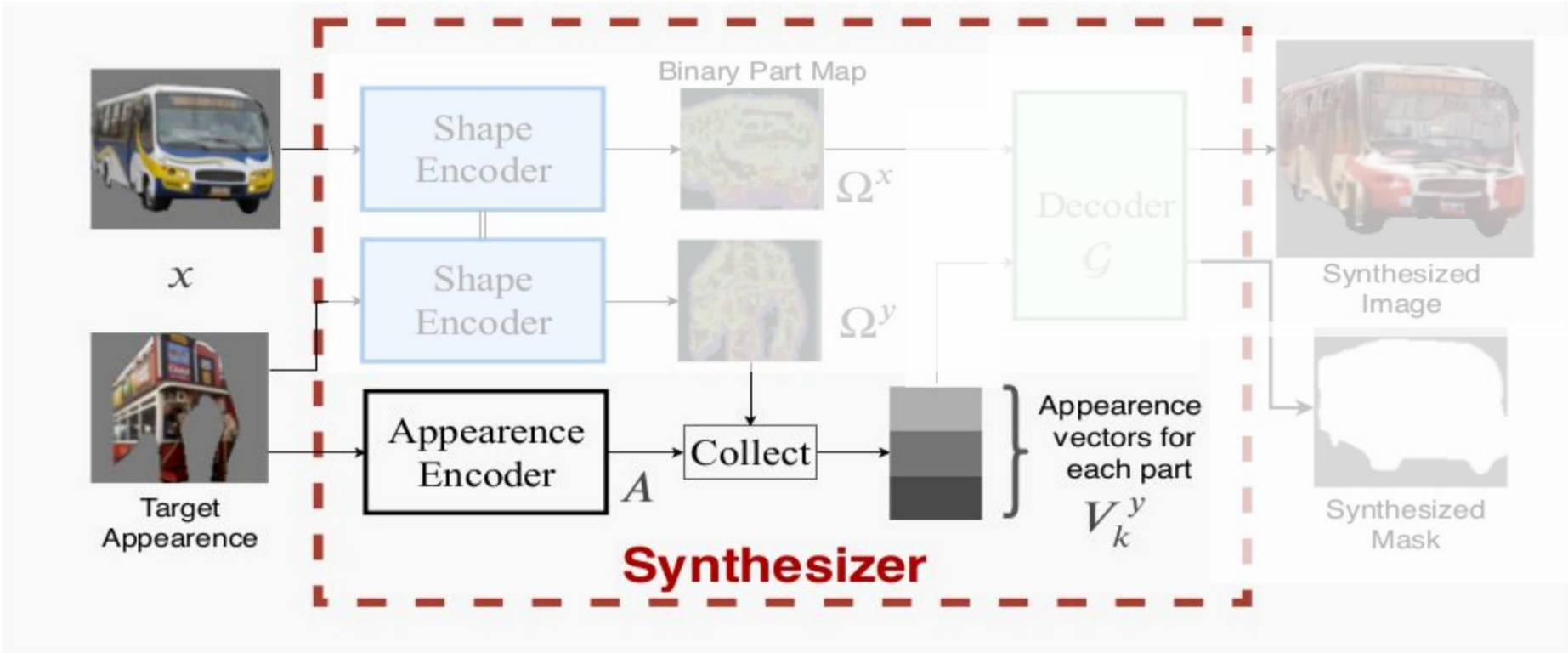
Encoding | **Decoding**

Inside the synthesizer network



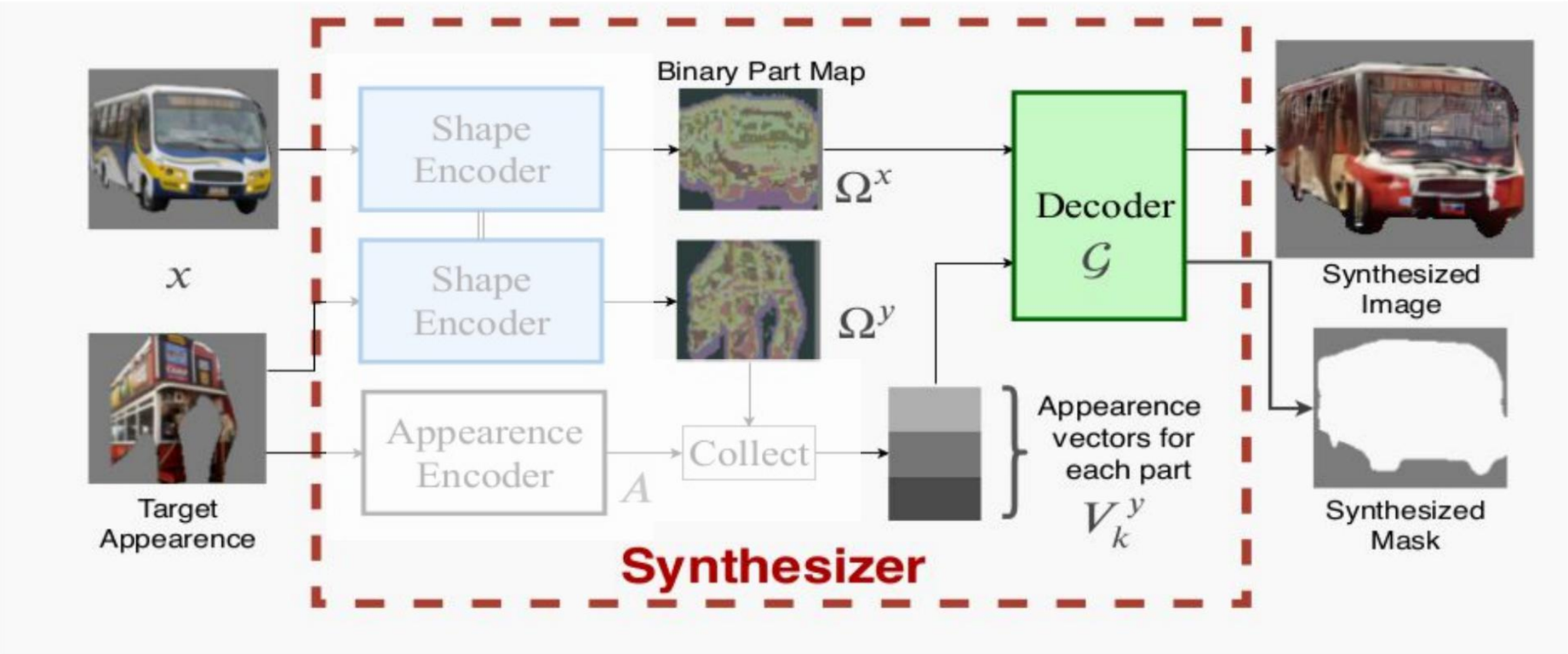
Shape Encoder $\longrightarrow$ Part segmentation bottleneck to allow only spatial information

Inside the synthesizer network



Appearance Encoder → one vector per part without spatial dimensions

Inside the synthesizer network

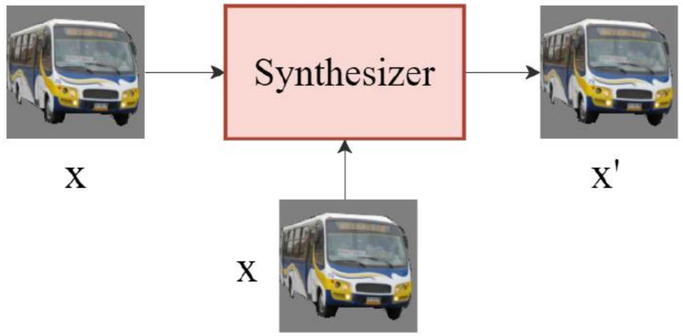


Decoder combines the shape and appearance codes to generate the object

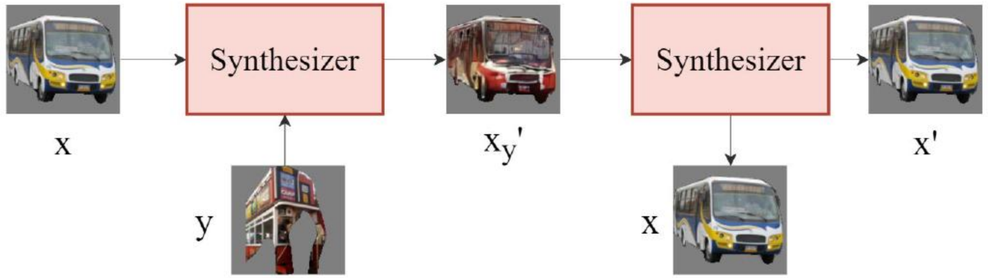
Training the Synthesizer

Reconstruction based

a. Auto encoding

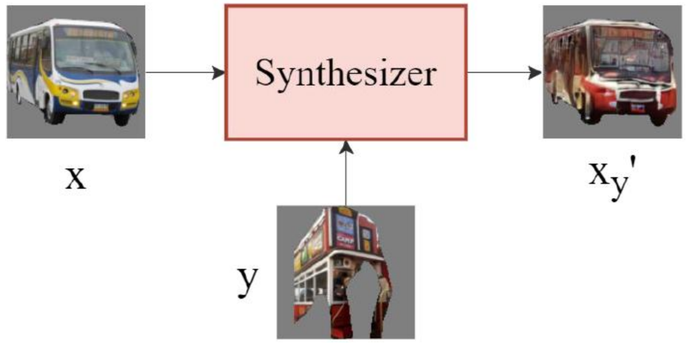


b. Cyclic Auto encoding

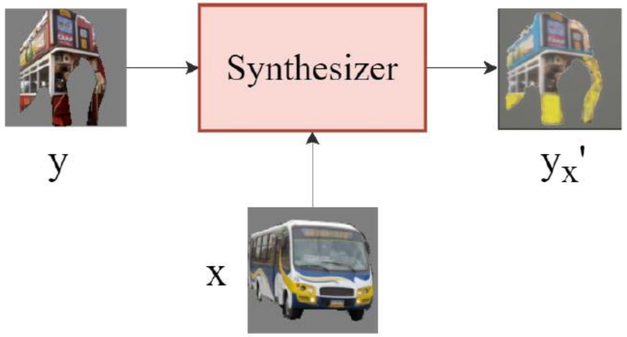


GAN based

c. Cross encoding

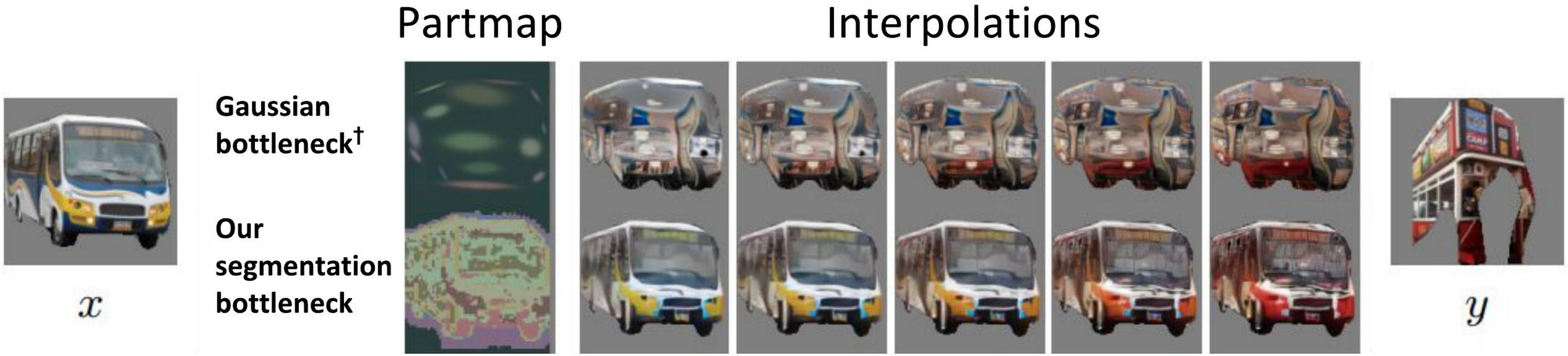


d. Cross encoding



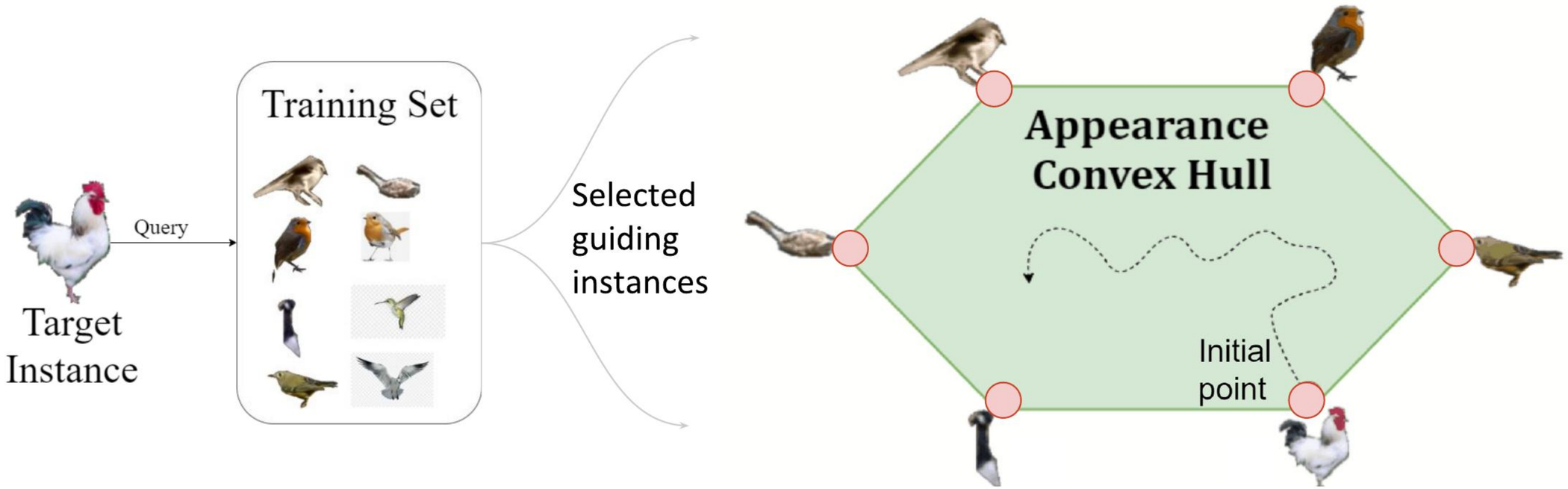
Semantic Adversaries with the Synthesizer

Synthesizer enables appearance interpolation



†Applied in prior works (Lorenz et al. CVPR19', Siarohin et al. CVPR19')

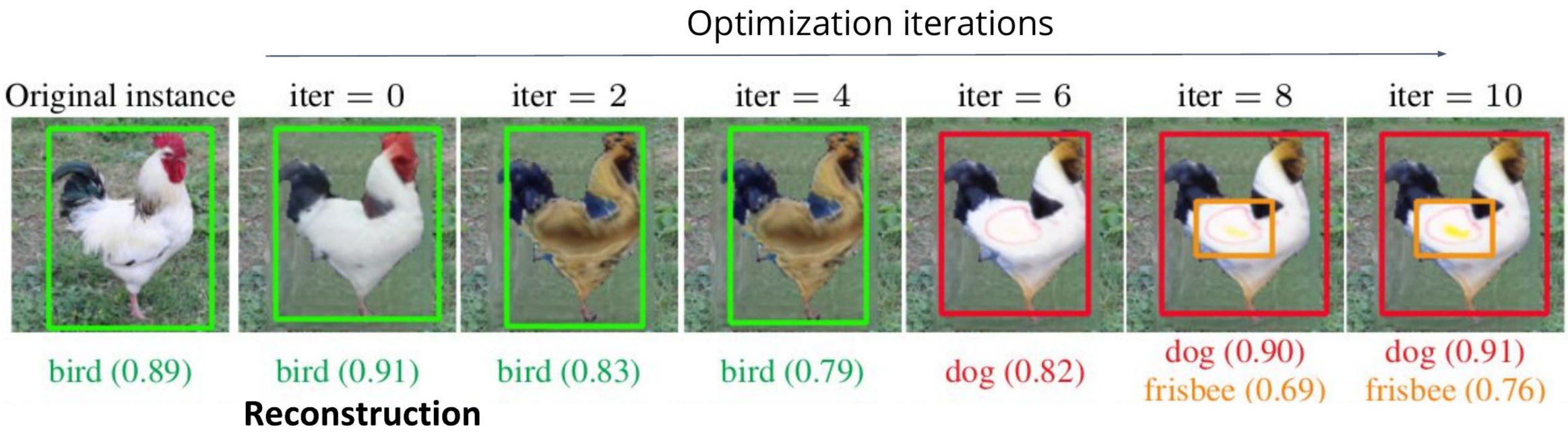
Constrained adversarial optimization



$$\alpha_k^j = \mathrm{softmax}_n a_k^j$$

$$\mathcal{V}_k^{adv} = \left[\sum_{j=1}^{n} \alpha_1^j V_1^{i_j} \cdots \sum_{j=1}^{n} \alpha_k^j V_k^{i_j} \right]$$

Illustration of the semantic adversarial attack



Evaluating semantic adversaries

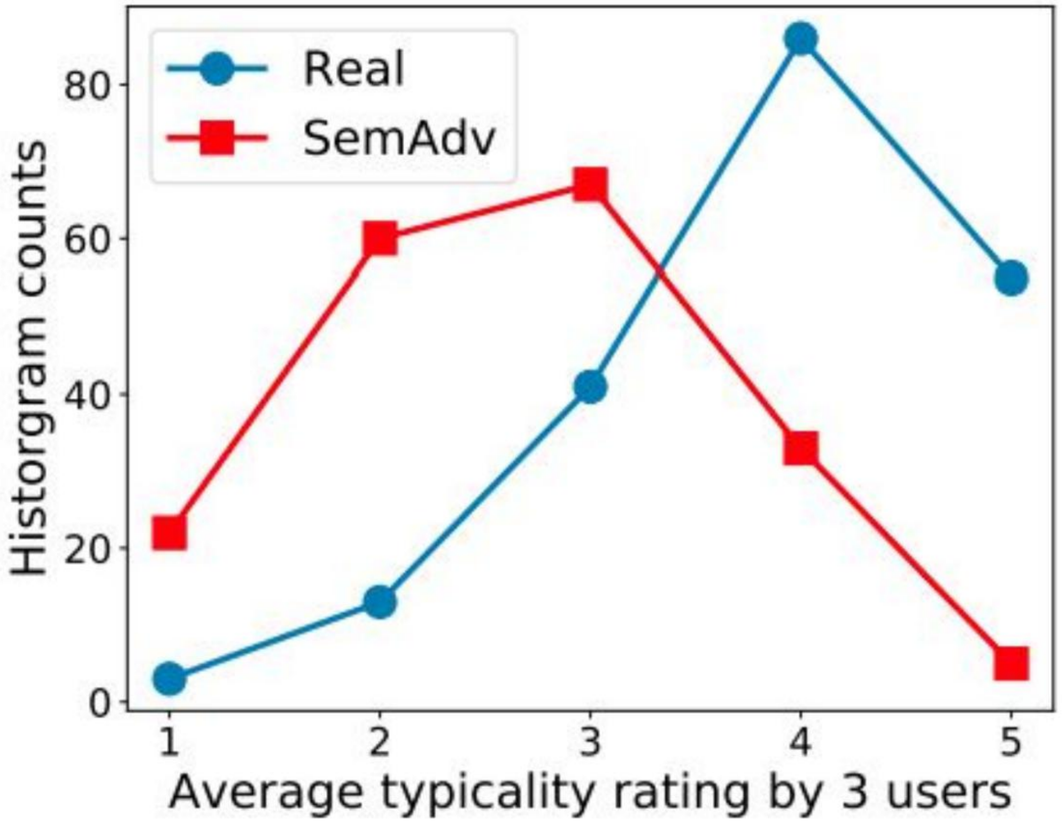
Quantitative effectiveness of the attack

Testing data	**Attack on**	**No. of objects**	**mAP@ 0.5 ↓**
Original data	-	-	81.2
L_∞ adversary (ϵ=8/255)	pixels	1	58.7
Semantic adversary	Appearance	1	62.4
Semantic adversary	Position + Appearance	1	59.5
Semantic adversary	Position + Appearance	2	46.5

Human study results - Class consistency

Testing data	mAP@ 0.5 of the detector
Original data	99%
Randomized label	11%
Semantic adversary	**93%**

Human study results - Typicality



More unusual appearance

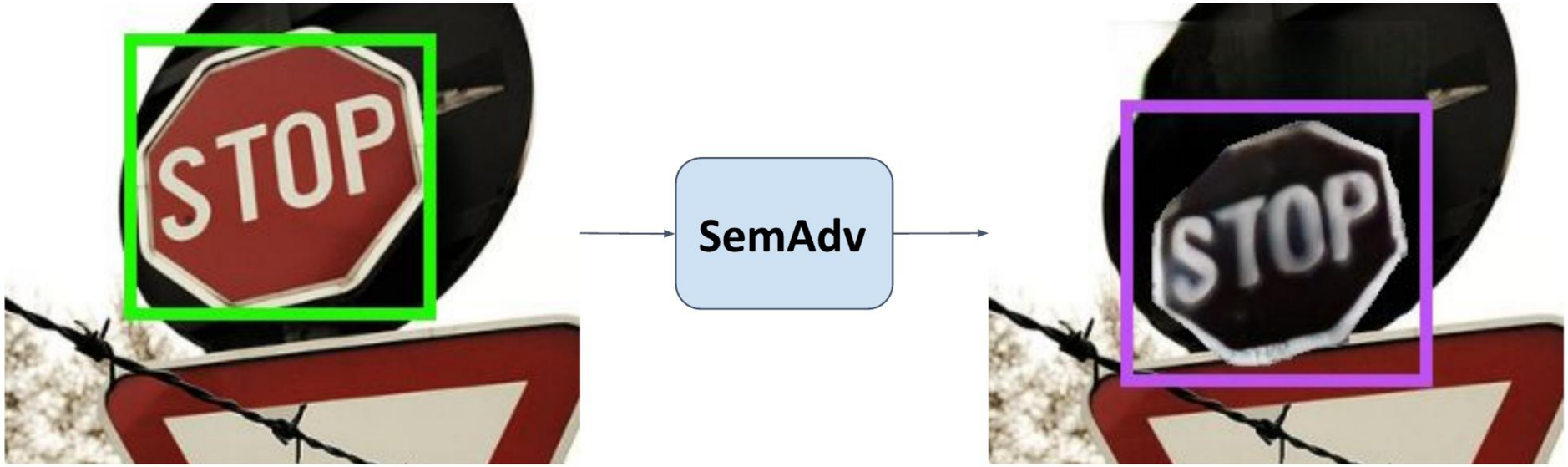
Data augmentation results: Summary

- Small but consistent improvement on **three datasets**
- Larger gains on out-of-dataset distribution test samples

	IID test set	**OOD test set**
COCO	+ 2.17%	**+ 4.6%** (on Unrel)
PASCAL VOC	+ 1.35%	**+ 4.9%** (on Unrel)
BDD 100k	+ 1.38%	+ 1.15% (on D2-city)

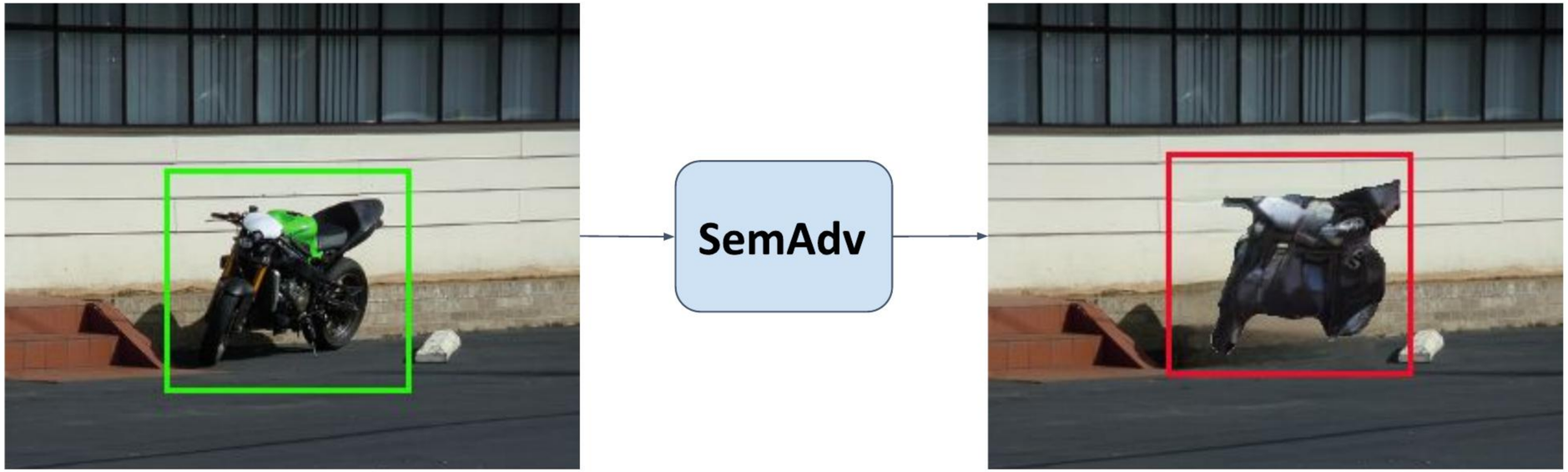
Qualitative failure modes

Synthesized hard examples - Camouflaging



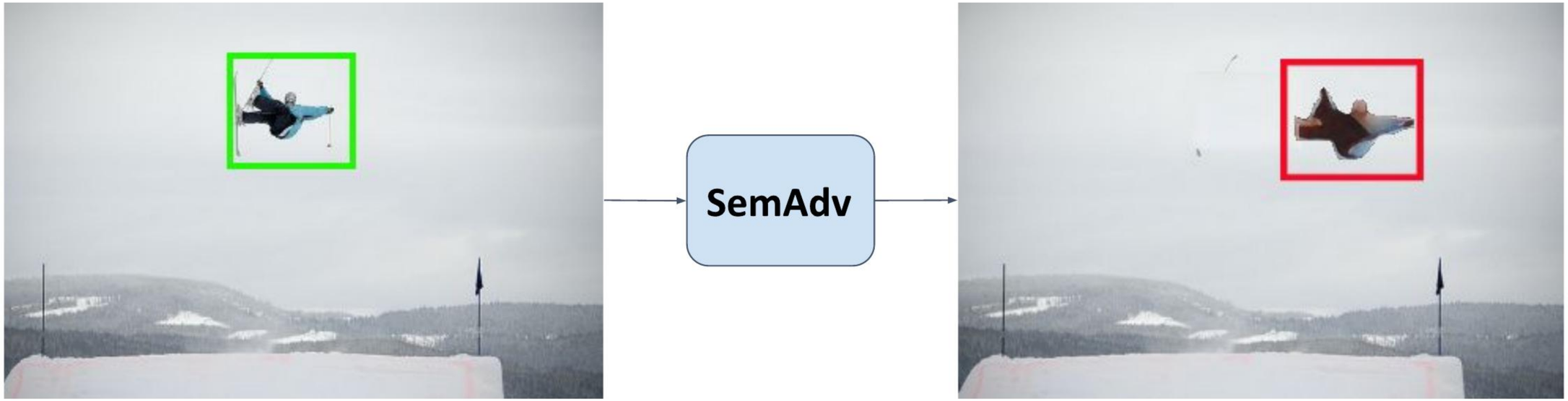
Prediction : **Stop Sign** ✔ **No detection** ❌

Synthesized hard examples - Appearance



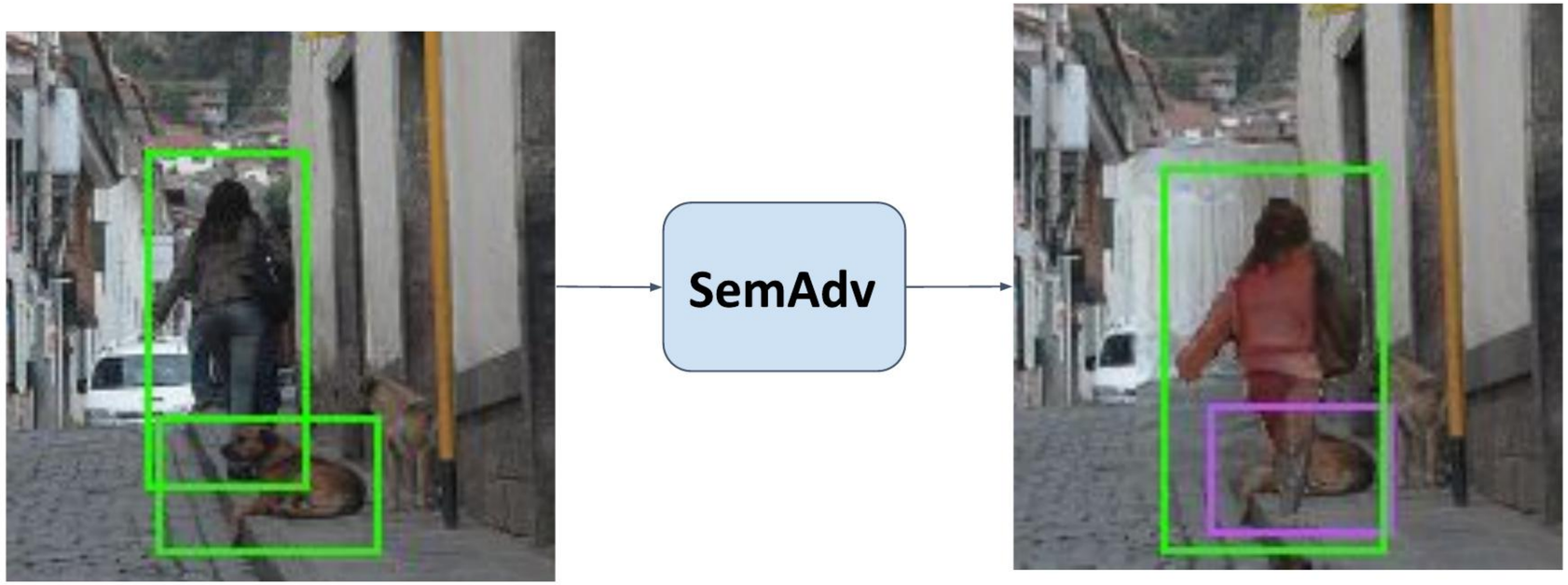
Prediction : **Motorcycle** ✔ **Backpack** ✘

Synthesized hard examples - Context



Prediction : **Person** ✔ **Airplane** ✘

Synthesized hard examples - Occlusion



Prediction : Dog ✓ Person ✓

No detection ✗ Person ✓

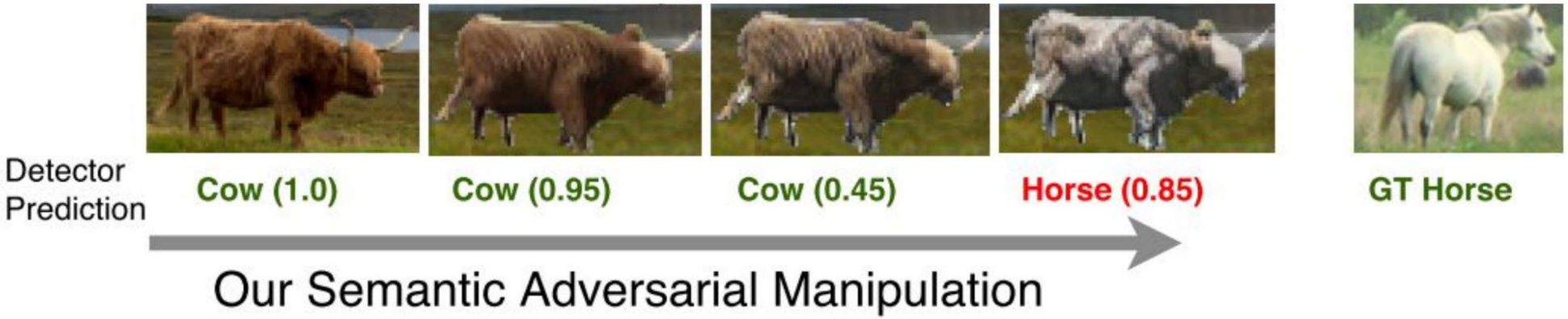
More examples in the paper & the supplementary

Takeaways

Summary

- A new way to test computer vision models through semantic adversarial data synthesis
- A disentangled synthesizer + constrained adversarial optimization = hard examples with large appearance changes, yet plausible to human eye
- Finds novel failure modes of the detector and helps improve robustness

Semantic adversaries
A new way to test computer vision models



Code available at https://github.com/rakshithShetty/SemanticAdversary

Thank you for listening