EditGAN: High-Precision Semantic Image Editing

Huan Ling, Karsten Kreis et al.

NVIDIA, Univ. of Toronto, Vector Institute, and MIT

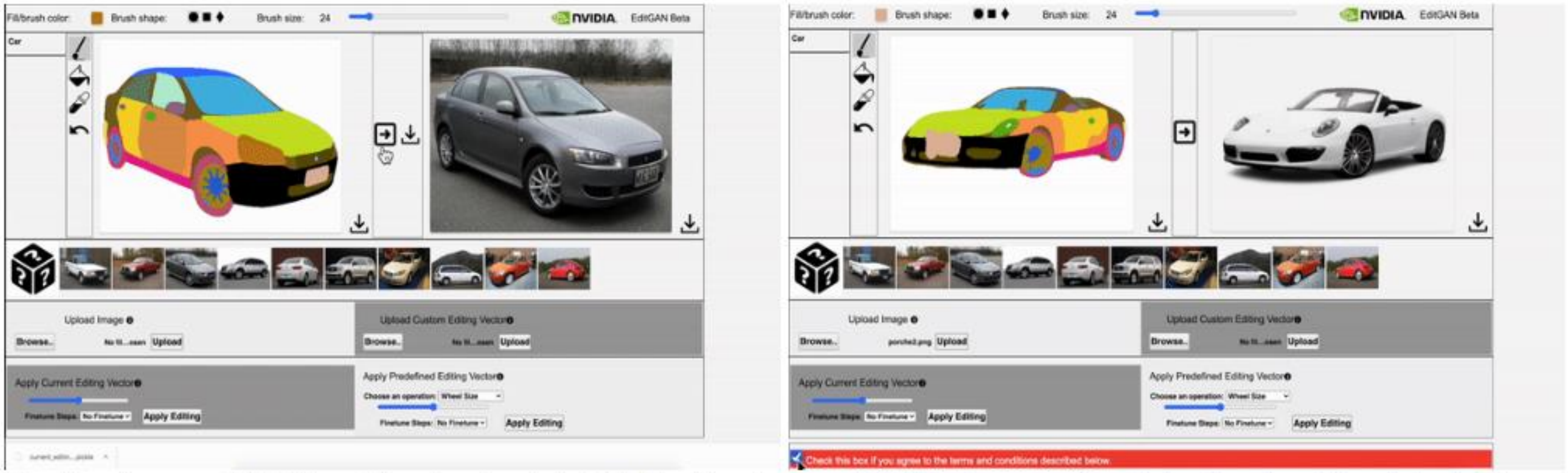
NIPS 2021

Presented by Minho Park

Demos

- https://nv-tlabs.github.io/editGAN/

Demos



Left: The video showcases EditGAN in an interacitve demo tool. *Right:* The video showcases EditGAN where we apply multiple edits and exploit pre-defined editing vectors.

Contributions

- EditGAN is the first GAN-driven image editing framework which simultaneously offers very high-precision editing (Method / Qualitative Results).
- EditGAN requires only very little annotated training data and does not rely on external classifiers (Method).
- EditGAN can be run interactively in real time (Qualitative Results).
- EditGAN works on real embedded, GAN-generated, and even out-of-domain images (Additional Results).

Related Work

- SemanticGAN

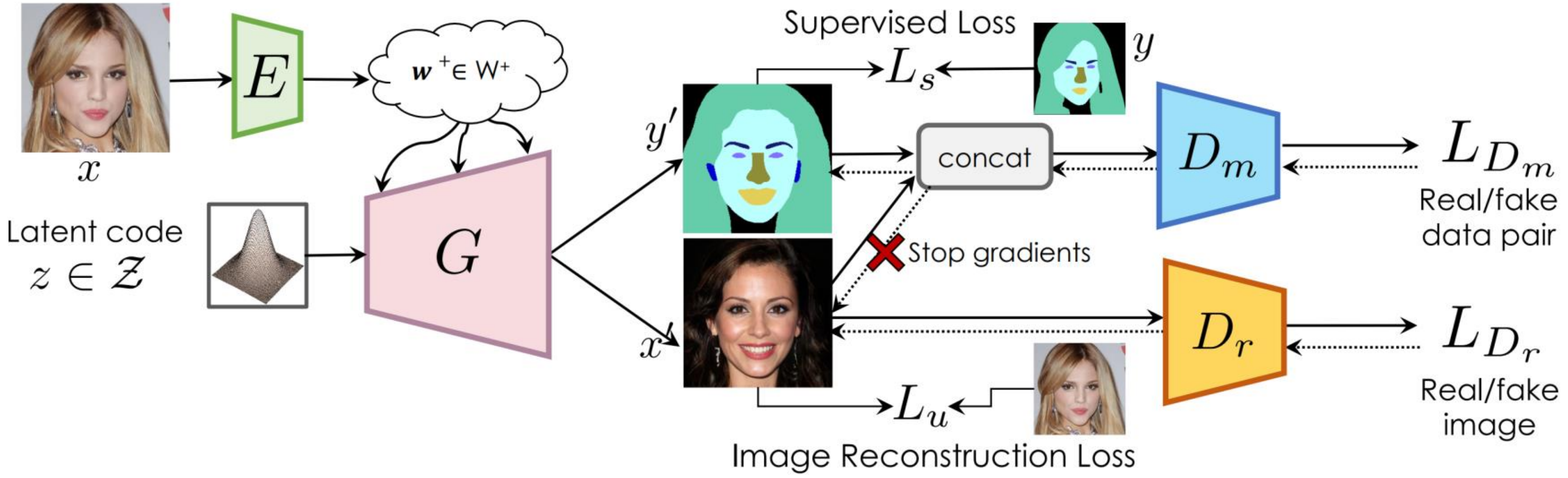


Figure 2: **Model Overview.** Generator G and discriminators D_m and D_r are trained with adversarial objectives $\mathcal{L}_G$ (not indicated here), $\mathcal{L}_{D_m}$ and $\mathcal{L}_{D_r}$. We do not backpropagate gradients from D_m into the generator's image synthesis branch. We train an additional encoder E in a supervised fashion using image and mask reconstruction losses $\mathcal{L}_u$ and $\mathcal{L}_s$.

Li, Daiqing, et al. "Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

Related Work

- SemanticGAN

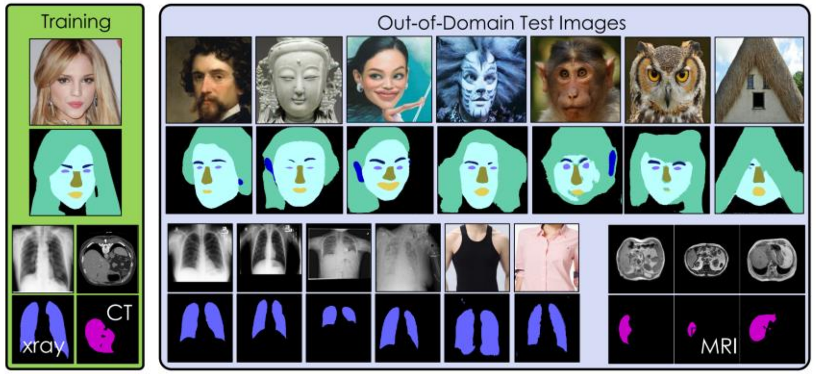


Figure 1: **Out-of-domain Generalization.** Our model trained on real faces generalizes to paintings, sculptures, cartoons and even outputs plausible segmentations for animal faces. When trained on chest x-rays, it generalizes to multiple hospitals, and even hallucinates lungs under clothed people. Our model also generalizes well from CT to MRI medical scans.

Figure 8: **Extreme Out-Of-Domain Segmentation.** Results on images with a large visual gap to CelebA, on which our model was trained.

Overview

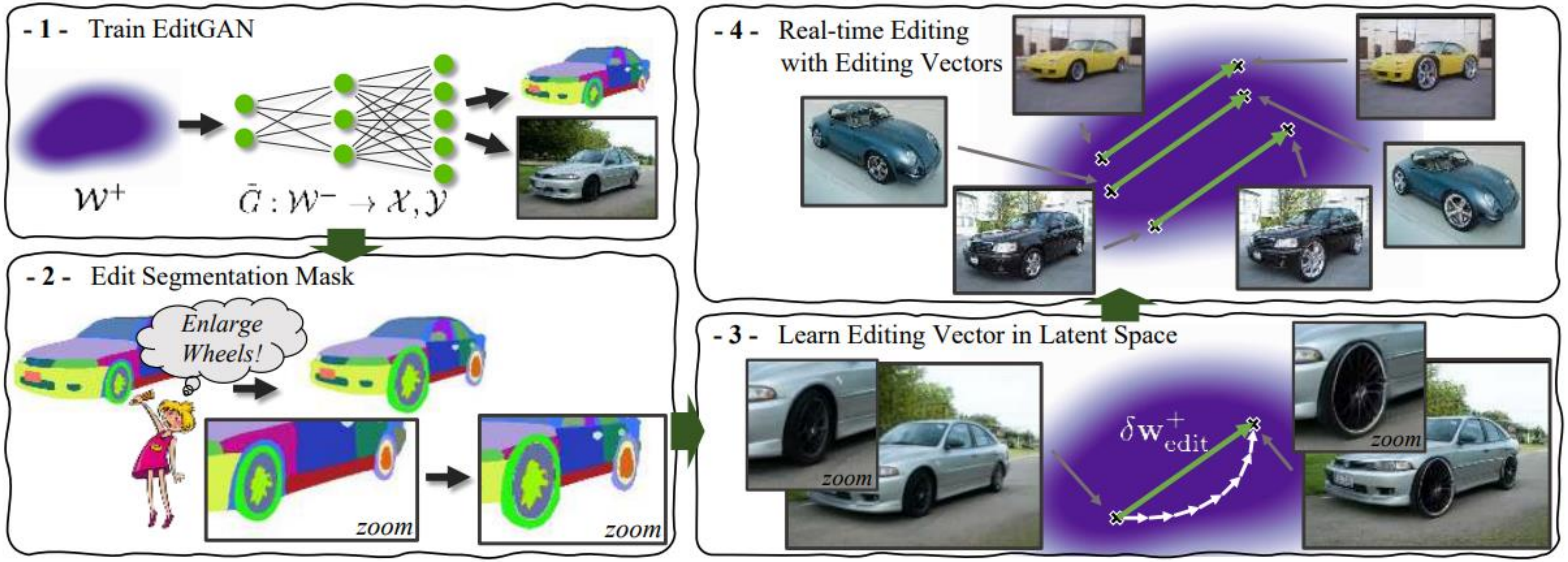
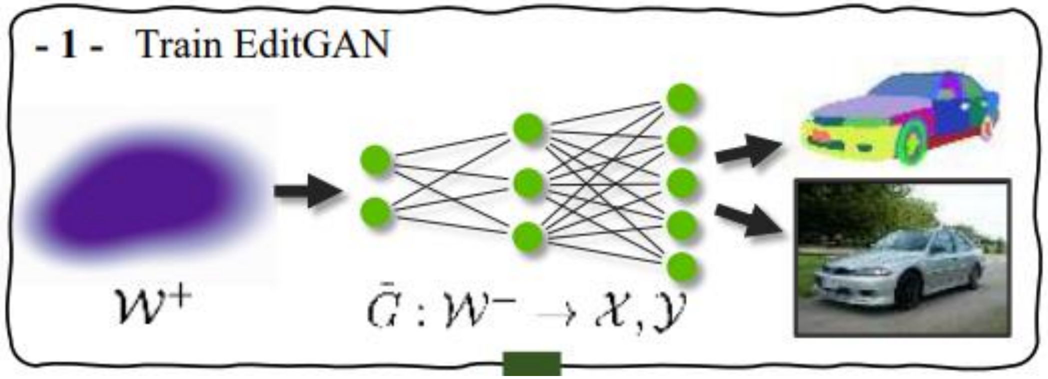


Figure 2: (**1**) EditGAN builds on a GAN framework that jointly models images and their semantic segmentations. (**2 & 3**) Users can modify segmentation masks, based on which we perform optimization in the GAN's latent space to realize the edit. (**4**) Users can perform editing simply by applying previously learnt editing vectors and manipulate images at interactive rates.

Method

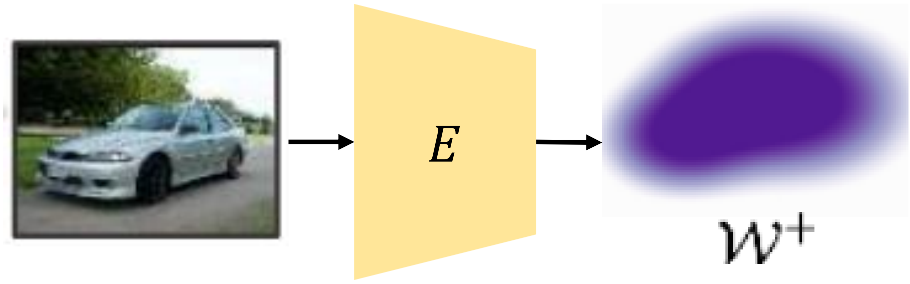
- **Train EditGAN.**
- Learn a joint distribution $p(x, y)$ over images x and pixel-wise semantic segmentation labels y (i.e., SemanticGAN, DatasetGAN).
- Both methods model $p(x, y)$ by adding **an additional segmentation branch** to the image generator, which is a pre-trained StyleGAN (EditGAN follows DatasetGAN).



Zhang, Yuxuan, et al. "Datasetgan: Efficient labeled data factory with minimal human effort." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
Li, Daiqing, et al. "Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

Method

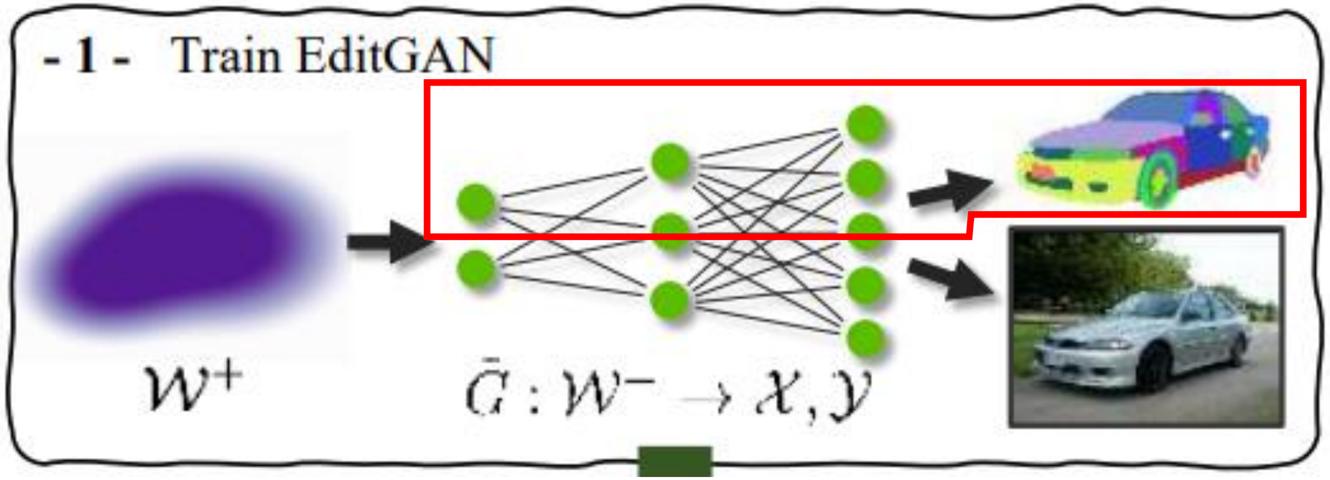
- **Train EditGAN.**
- Pretrained StyleGAN2
- GAN inversion: Train an encoder that embeds images into W^+ space.
  - Objective function: L2, LPIPS
  - Explicitly regularize the encoder with the known underlying latent codes.



Inversion encoder

Method

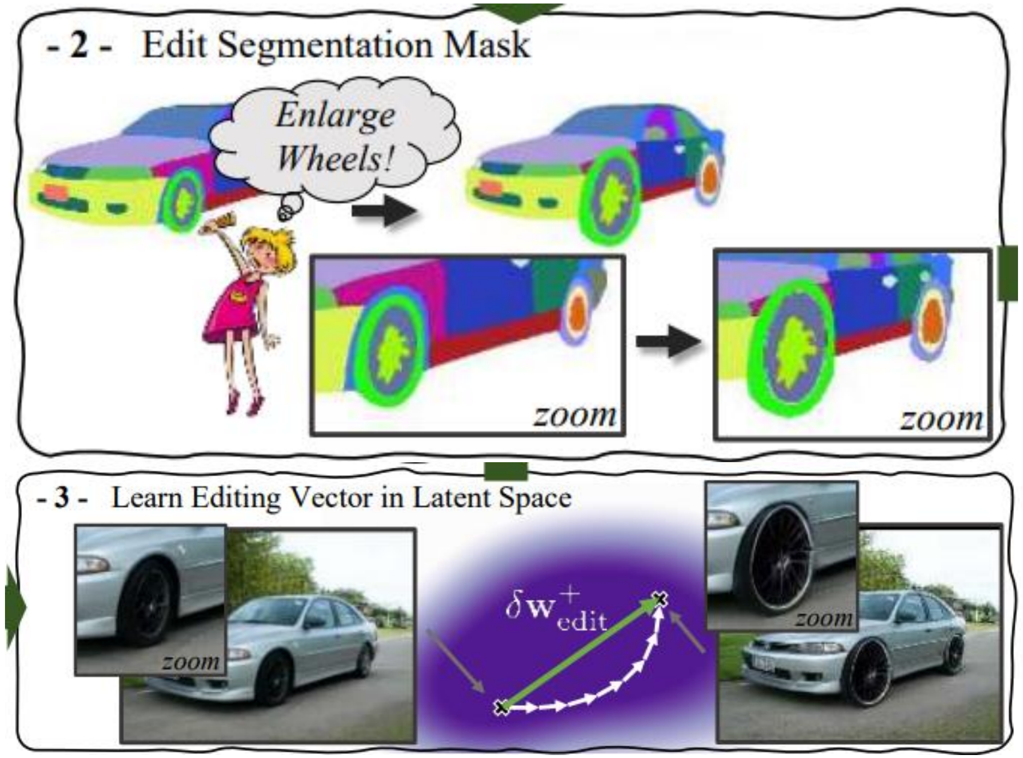
- **Train EditGAN.**
- Train the segmentation branch of the generator using the cross-entropy loss.

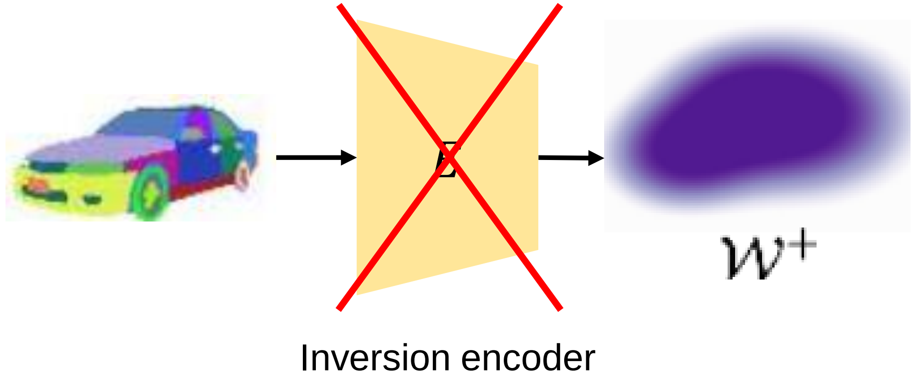


CrossEntropyLoss

Method

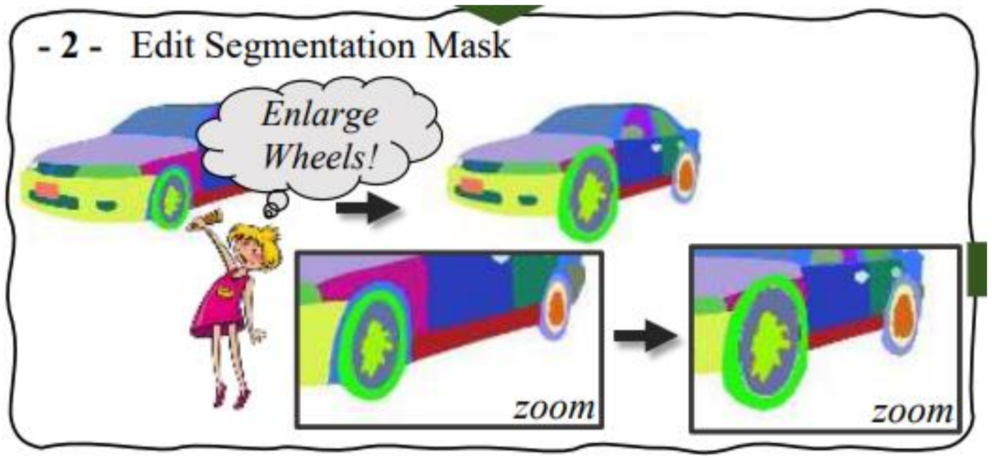
- **Learn editing vector in latent space.**
- Cannot use GAN inversion in segmentation label.
  I.e., Does not work well.





Method

- **Learn editing vector in latent space.**
- Optimization in latent space!
- Given an image to be edit: $x_{orig} \to w_{orig}^{+} \to (x_{orig}, y_{orig})$.
- Manually edit the segmentation map $y_{orig} \to y_{edit}$.
- Find w_{edit}^{+}, which consistent with y_{edit}.



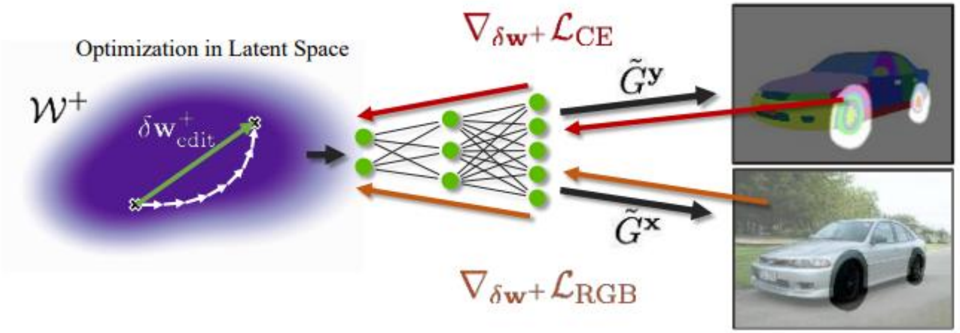


Figure 3: We modify semantic segmentations and optimize the shared latent code for consistency with the new segmentation *within* the editing region, and with the RGB appearance *outside* the editing region. Corresponding gradients are backpropagated through the shared generator. The result is a latent space editing vector $\delta \mathbf{w}_{edit}^{+}$.

Method

- **Learn editing vector in latent space.**
- Find an editing vector $\delta w^{+}_{edit} \in \mathcal{W}^{+}$, such that $(x_{edit}, y_{edit}) = G_{fixed}(w^{+} + \delta w^{+})$.
- The region of interest is formally given by $r = \underbrace{\{p | c^{y}_{p} \in Q_{edit}\}}_{\text{will be removed}} \cup \underbrace{\{p | c^{y_{edit}}_{p} \in Q_{edit}\}}_{\text{will be added}}$.
  - Q_{edit}: edit-specific pre-specified list.
  - E.g., if we edit wheels, Q_{edit} would contain all part labels related to the wheels.

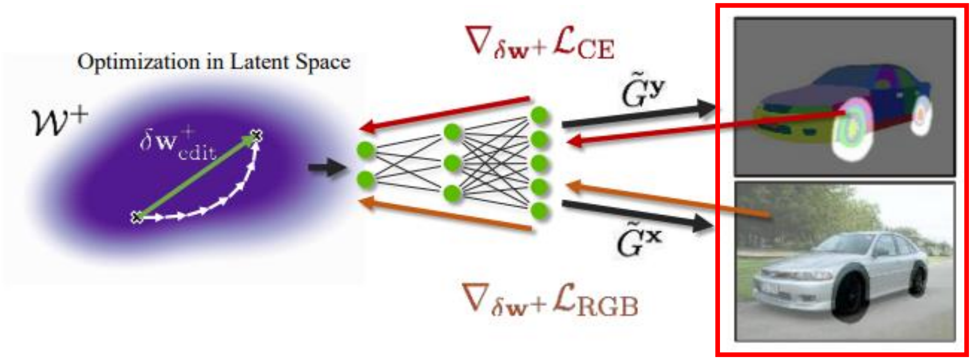


Figure 3: We modify semantic segmentations and optimize the shared latent code for consistency with the new segmentation *within* the editing region, and with the RGB appearance *outside* the editing region. Corresponding gradients are backpropagated through the shared generator. The result is a latent space editing vector $\delta \mathbf{w}^{+}_{edit}$.

Method

- **Learn editing vector in latent space.**
- Find an editing vector $\delta w^{+}_{edit} \in \mathcal{W}^{+}$, such that $(x_{edit}, y_{edit}) = G_{fixed}(w^{+} + \delta w^{+})$.
- Let $r = \{p | c_p^y \in Q_{edit}\} \cup \{p | c_p^{y_{edit}} \in Q_{edit}\}$.

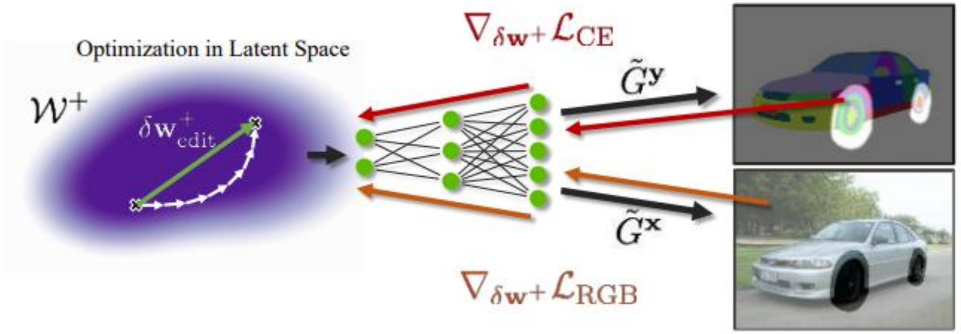


Figure 3: We modify semantic segmentations and optimize the shared latent code for consistency with the new segmentation *within* the editing region, and with the RGB appearance *outside* the editing region. Corresponding gradients are backpropagated through the shared generator. The result is a latent space editing vector $\delta \mathbf{w}^+_{\text{edit}}$.

$$\mathcal{L}_{\text{RGB}}(\delta \mathbf{w}^+) = L_{\text{LPIPS}}(\tilde{G}^{\mathbf{x}}(\mathbf{w}^+ + \delta \mathbf{w}^+) \odot (1 - r),\ \mathbf{x} \odot (1 - r)) + L_{L2}(\tilde{G}^{\mathbf{x}}(\mathbf{w}^+ + \delta \mathbf{w}^+) \odot (1 - r),\ \mathbf{x} \odot (1 - r))$$

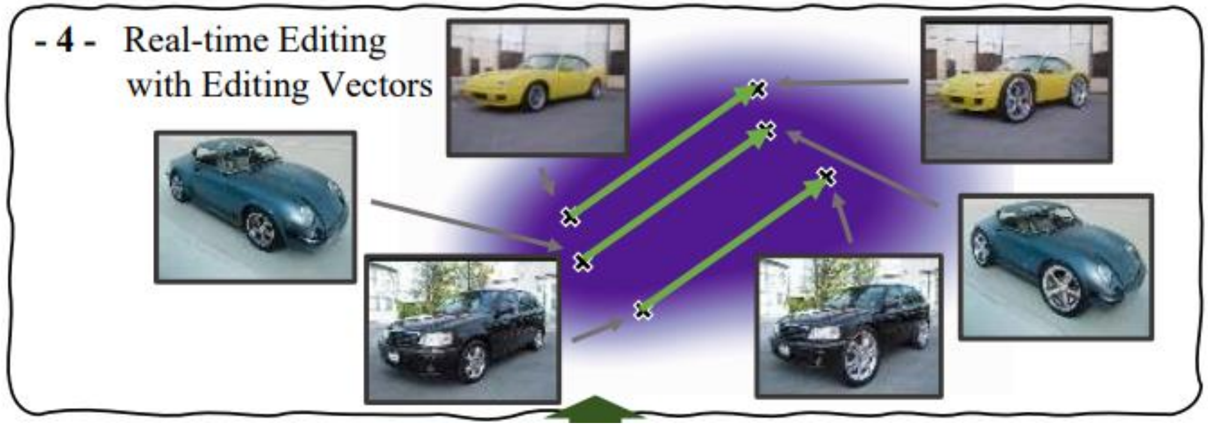
$$\mathcal{L}_{\text{CE}}(\delta \mathbf{w}^+) = H(\tilde{G}^{\mathbf{y}}(\mathbf{w}^+ + \delta \mathbf{w}^+) \odot r,\ \mathbf{y}_{\text{edited}} \odot r)$$

$$\mathcal{L}_{\text{ID}}(\delta \mathbf{w}^+) = \langle R(\tilde{G}^{\mathbf{x}}(\mathbf{w}^+ + \delta \mathbf{w}^+)), R(\mathbf{x}) \rangle$$ Just for human face

pretrained ArcFace feature extraction network

Method

- δw_{edit}^{+} are semantically meaningful and **often disentangled** with other attributes (not always).
- $(x', y') = G(w^{+} + s_{edit}\delta w_{edit}^{+})$.
- **Image editing with EditGAN in three different modes:**
  - Real-time Editing with Editing Vector: editing vector only.
  - Vector-based Editing with Self-Supervised Refinement: editing vector + additional optimization at test time.
  - Optimization-based Editing: optimization only.



Qualitative Results

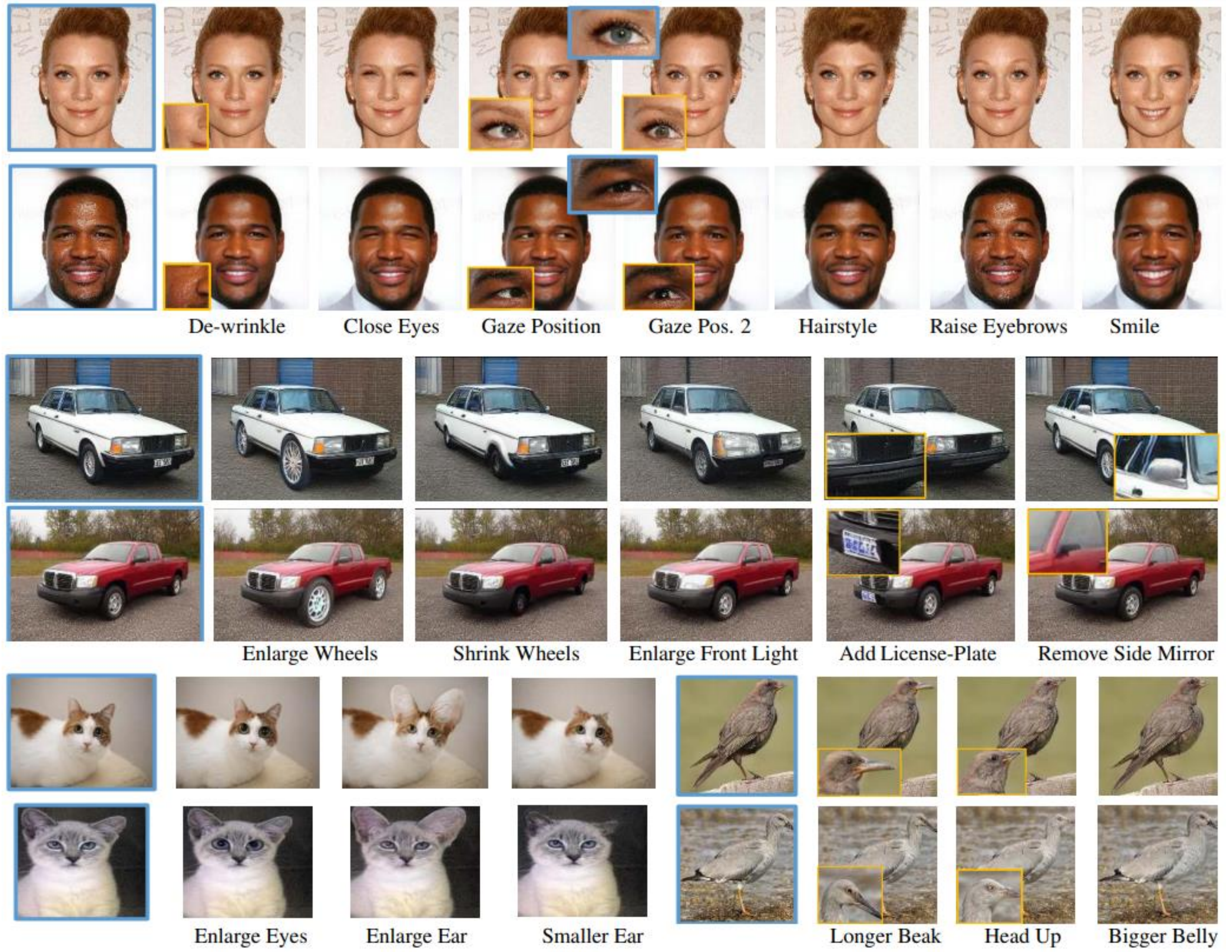


Figure 4: Examples of segmentation-driven edits with EditGAN. Results are based on editing with editing vectors and 30 steps self-supervised refinement. *Blue boxes*: Original images. *Orange boxes*: Zoom-in views.

Qualitative Results

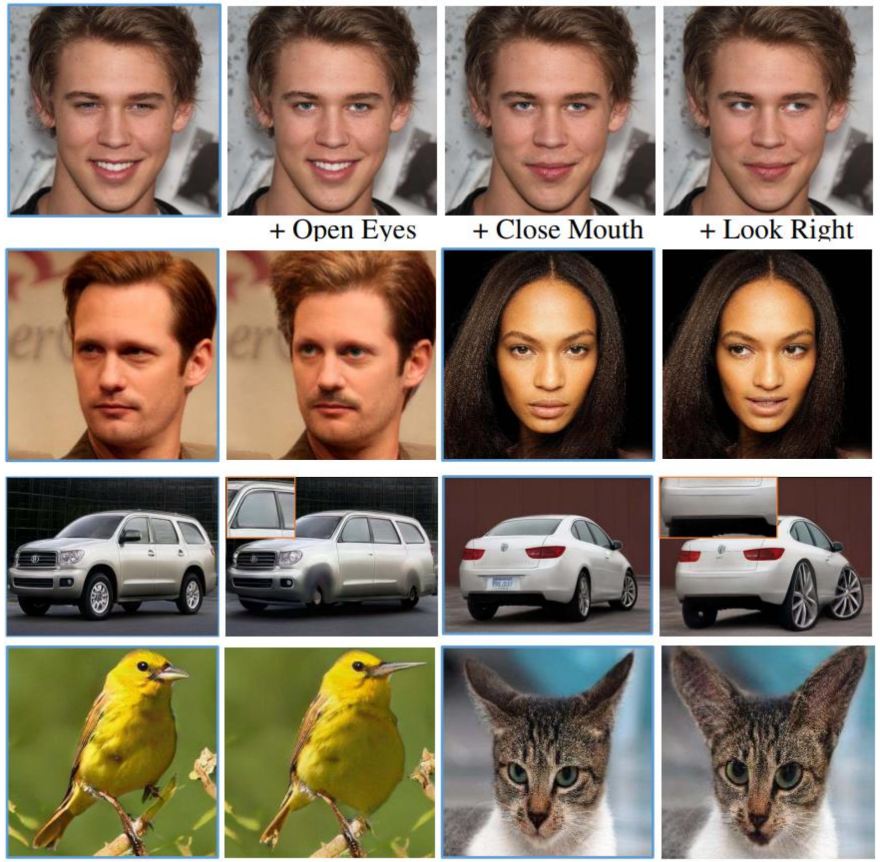


Figure 5: We combine multiple edits. Results are based on editing with editing vectors and 30 steps self-supervised refinement. *Blue boxes*: Original images. Edits in detail: *Second row, first person:* open eyes, add hair, add mustache. *Second person:* smile, look left. *Third row, first car:* remove mirror, remove door handle, shrink wheels. *Second car:* remove license plate, enlarge wheels. *Third row, bird:* longer beak, bigger belly, head up. *Third row, cat:* open mouth, bigger ear, bigger eyes.

Quantitative Results

Metric	# Mask Annot.	# Attribute Annot.	Attribute Acc.(%) ↑	FID ↓	KID ↓	ID Score ↑
MaskGAN [10]	30,000	-	77.3	46.84	0.020	0.4611
LocalEditing [18]	-	-	26.0	41.26	0.012	0.5823
LocalEditing - Encoding4Editing [81]	-	-	41.75	48.28	0.016	0.6603
InterFaceGAN [13]	-	30,000	83.5	**39.42**	**0.010**	0.7295
EditGAN (ours)	16	-	**91.5**	41.74	0.013	0.7047
EditGAN$^+$30 (ours)	16	-	85.8	40.83	0.012	**0.7452**
StyleGAN2 Distillation [82]	-	30,000	98.3	45.09	0.013	0.7823

Table 1: Quantitative comparisons to multiple baselines on the smile edit benchmark.

Out-of-domain Results

Figure 6: We combine multiple edits on out-of-domain images. Results are based on editing with editing vectors and 30 steps self-supervised refinement. Edits in detail: *First row, first example:* look left, frown. *Second example:* smile, look right. *Second row, first example:* open eyes, lift eyebrow. *Second example:* open eyes.

Limitations

- EditGAN is limited to images that can be modeled by the GAN (same as semanticGAN).
- Optimization is needed in inference time.
  - Optimization only: 11.4s.
  - Self-supervised refinement: 4.2s.
- Each editing vector δw_{edit}^{+} is not always disentangled.