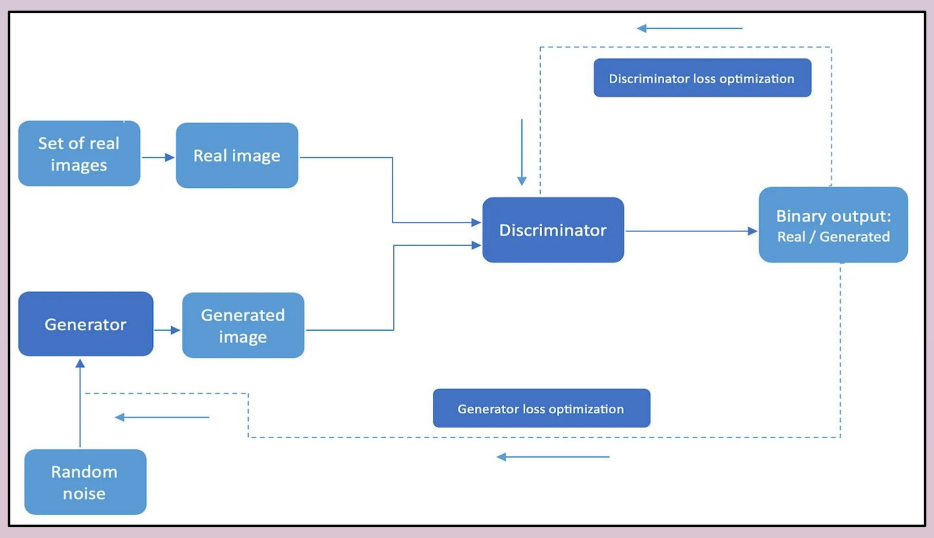
Exploring Latent Space of GAN

Hetvi Patel, Kevin Shah, Prajwal Singh, Shanmuganathan Raman
CVIG Lab, Indian Institute of Technology Gandhinagar

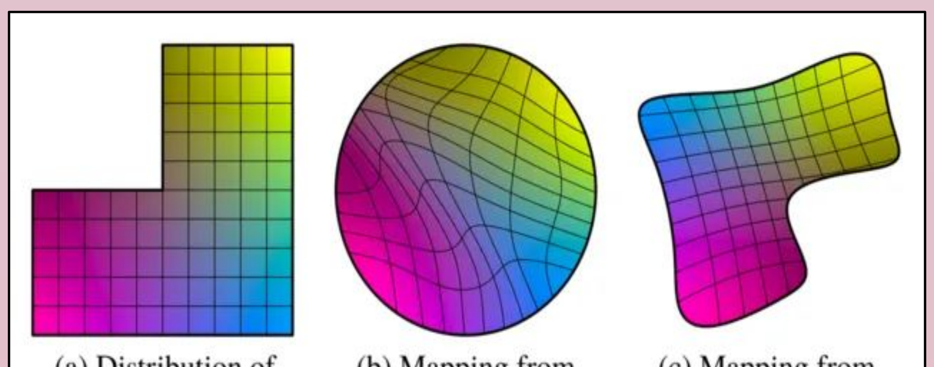
PROBLEM STATEMENT

- Investigate methods for **manipulating GAN latent spaces** to achieve **precise modifications** in synthesized images
- Allow users with fine-grained **control** over **specific features** of generated images by enabling **direct** editing of the **latent space** representation.
- Explore how **GAN Inversion** concepts can be used in these techniques for **latent space editing**

LATENT SPACE of GANs



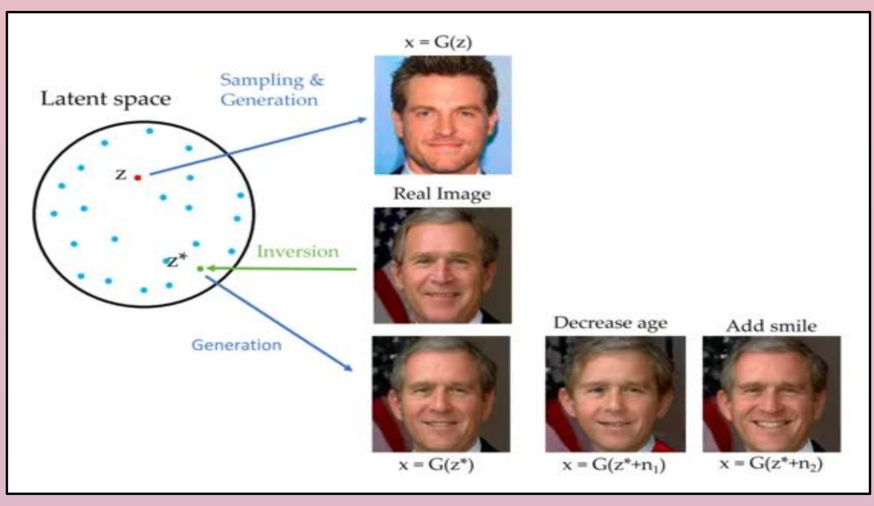
GAN Latent Space (**Z-space**):
- **Multi-dimensional space** where GANs map random noise input to generate realistic outputs.
- Encapsulates the **learned features** of the image data distribution.
- Operations like **interpolation** and **manipulation** to produce **semantically** meaningful changes in generated outputs.



(a) Distribution of features in training set (b) Mapping from $\mathcal{Z}$ to features (c) Mapping from $\mathcal{W}$ to features

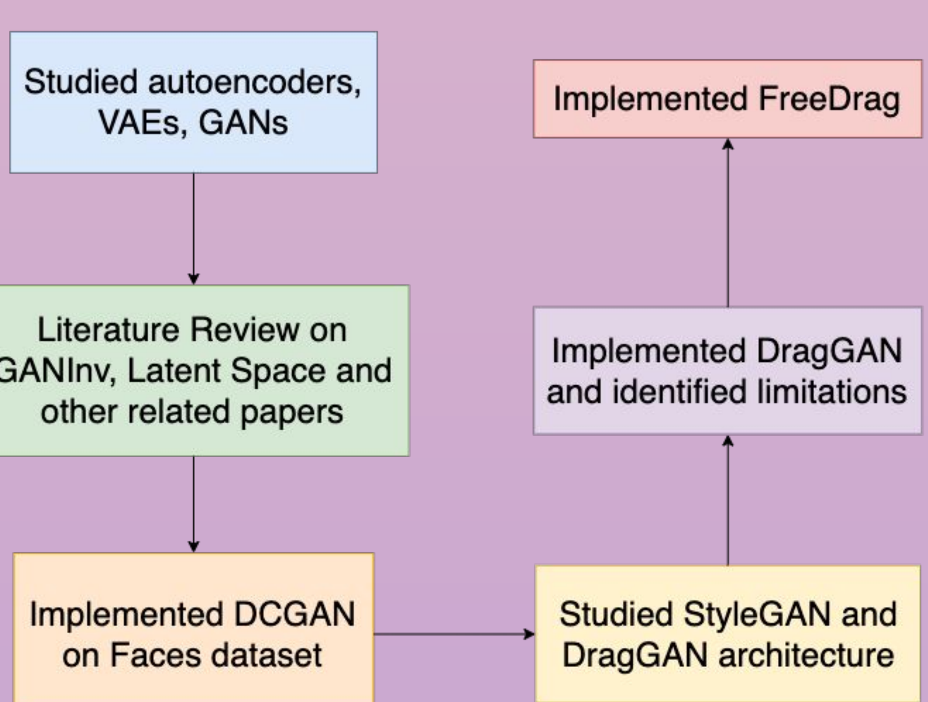
- **StyleGAN** introduced W space to better incorporate **semantic** information.
- Mapping of **Z space to W space** to obtain more **disentangled features** in the W space.

GAN INVERSION



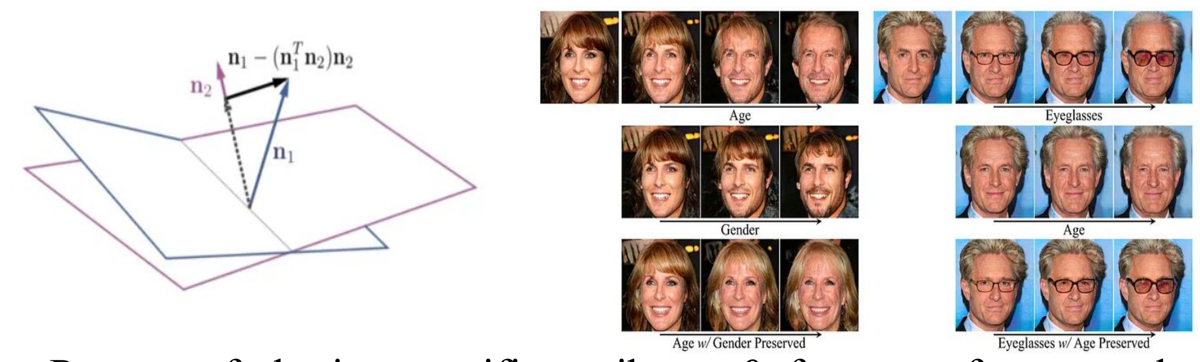
- **Problem**: GANs can synthesize high-quality fake images from random latent codes (z), but cannot infer latent codes from real images.
- GAN inversion finds a latent code (z*) that can reconstruct a given real image (x) through the GAN generator (G).

TIMELINE

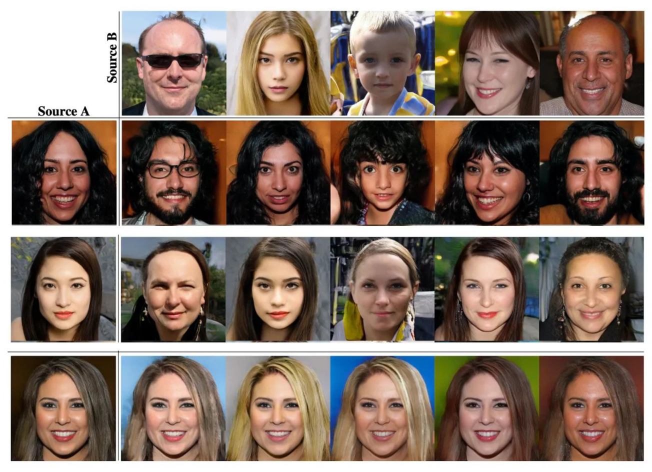


STYLE-GAN & DRAGGAN

StyleGAN: *Conditional Manipulation & Semantic Mapping*



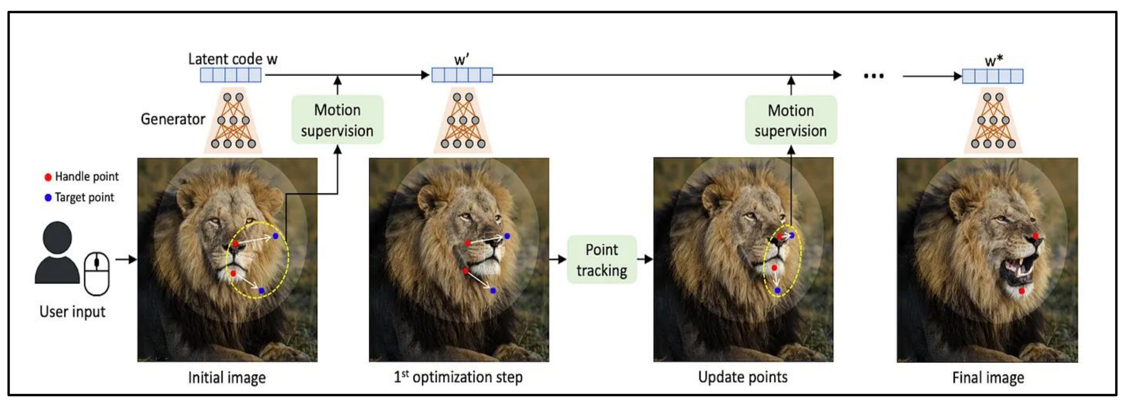
- Process of altering specific attributes & features of generated images without changing the other sematics of an image



StyleGAN showcasing style mixing

While **StyleGAN** introduced style mixing along with maintaining a great quality of generated images, *it doesn't provide users with complete and precise control over modifications* to be made. **DragGAN** offers exactly that — **interactive and complete control using a drag-and-drop methodology**.

Two-step Optimization process of DragGAN:



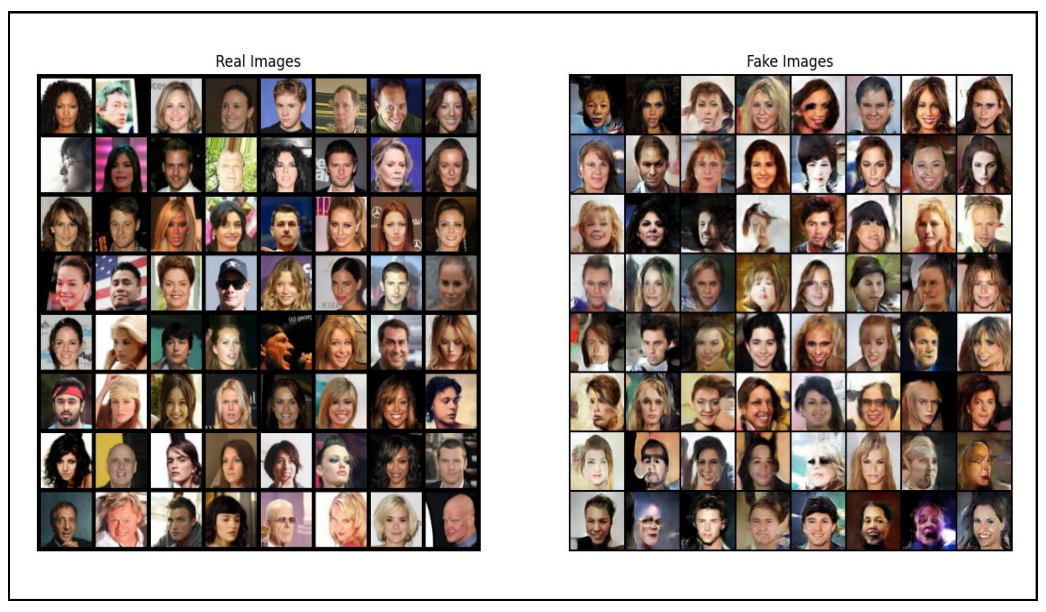
- Motion Supervision: First step involves *optimizing* for a loss function (**shifted patch loss** on GAN feature maps) that causes the handle points to move toward the target points.
- Point Tracking: Second step updates the handle points to track the object itself so that the handle points are updated after the motion supervision step, hence providing precise new locations of handle points. This step is achieved using **Nearest Neighbor Interpolation**

Red points = Handle points & Blue points = Target points

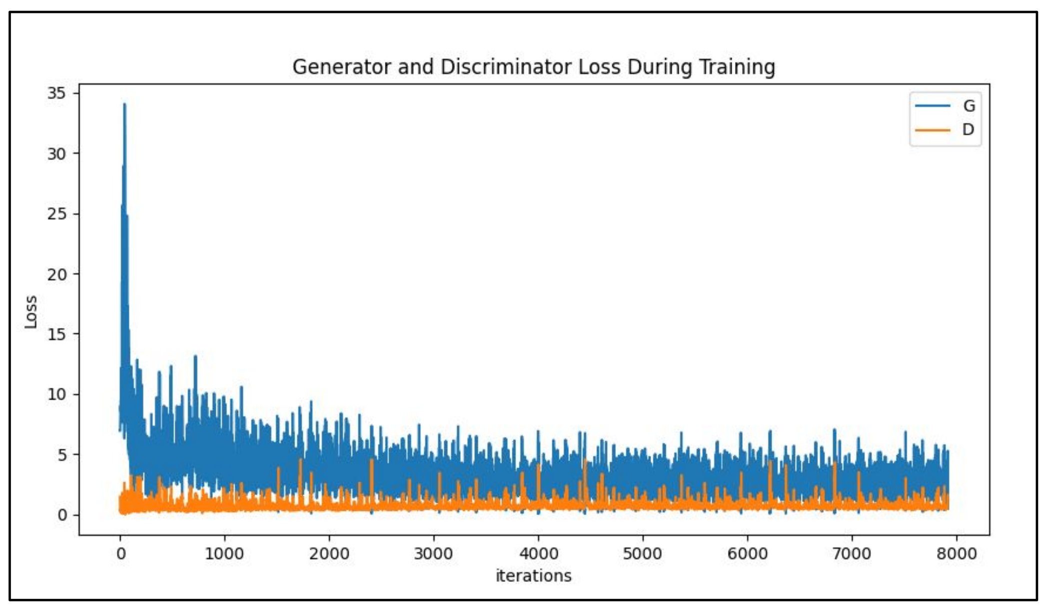
What does DragGAN do that other GANs haven't explored?
(Novel ways of generating GAN-based manipulated content):
- Interactive, point-based image manipulation
- Flexibility in generation
- Precision-driven image adjustments
- Object / Category Agnostic (independence)

EXPERIMENTAL RESULTS

DCGAN Implementation: Using Modular approach (pytorch) from scratch, and learnt the internal working of GANs.

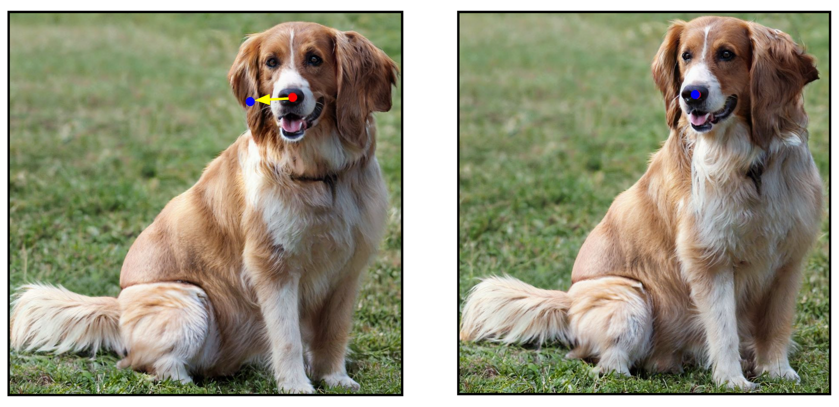


Fake images generated: DCGAN trained on Celeb-A Faces dataset



Variation in Training Loss of generator & discriminator of DCGAN

DragGAN vs FreeDrag: Limitations of DragGAN and FreeDrag's Enhanced version



DragGAN: *Change in the direction of face*

- FreeDrag: Two enhancements to DragGAN:
  1. **Template feature via adaptive updating**
  2. **Line search with backtracking**
- Together, these two enhancements lead to a higher-efficiency and more reliable semantic dragging.

FreeDrag: *Increase in facial mile*

Original Image DragGAN FreeDrag

DragGAN saturates before the handle point reaches the target point and changes the direction of whole image.
FreeDrag successfully changes only the direction of face and the handle point reaches the target point.

APPLICATIONS & FUTURE WORK

- **Applications**: Medical Imaging, Robotics, Virtual Reality, Film & Television, Business Marketing, etc
- **Limitations** of DragGAN: "**miss tracking**", inaccuracy in tracking predetermined handle points, and "**ambiguous tracking**", where tracked points are potentially positioned in wrong regions that closely resemble the handle points

- **Next Steps** in Research
1. Fine-tuning the model **FreeDrag** on **few-shot** (K-way N-shot learning) data so the model can **edit real world** custom images
2. Build upon FreeDRAG model to incorporate diverse variety of classes and also **include Out of Distribution** images

RÉFÉRENCES

- W. Xia, Y. Zhang, Y. Yang, J. -H. Xue, B. Zhou and M. -H. Yang, "GAN Inversion: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3121-3138, 1 March 2023
- Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. 2023
- Ling, P., Chen, L., Zhang, P., Chen, H., & Jin, Y. FreeDrag: Feature Dragging for Reliable Point-based Image Editing. 2023.
- Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." 2019.