NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Automated Image Generation

Hill Seah Wen Qi (U2121346H)

Project ID

CCDS24-0163

Project Supervisor

Associate Professor Lu Shijian

Examiner

Associate Professor Lin Guosheng

Introduction

Outline

- Research Scope
- Motivation
- Objectives

Research Scope

FYP Title → FYP Research Scope

Automated → Training-Free

- Automated Generation implies minimal human input
- Chosen training-free methods
- Training-free take automation further by removing the need for retraining on new data or conditions
- This increases practicality, reduces computational costs, and supports rapid adaptation to new generation tasks

Research Scope

FYP Title → FYP Research Scope

Image Generation → Multi-conditional Image Generation (Diffusion Models)

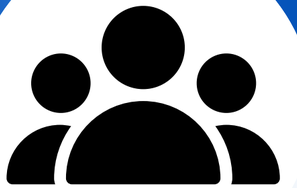
- Chosen multi-conditional generation, where multiple attributes or conditions guide the image output
- Chosen diffusion models as the primary image generation network as they offer strong controllability and high image quality
- Potential improvements or enhancements to existing problems and pain points

Motivation

Relevance of Training-Free Multi-Conditional Generation

"Generative design and content creation have the potential to unlock $150 billion to $275 billion in new value across industries like marketing, gaming, architecture, and product design."

- McKinsey & Company, 2023



"Generative models like Stable Diffusion have sparked a dramatic rise in visual content creation, with over 10 million images generated daily by users worldwide."

- Stanford University, 2024

Stable Diffusion 2.0

Training from scratch required approximately 150,000 GPU-hours using 256 NVIDIA A100 GPUs, culminating in a cost of around $600,000.

Motivation

Challenges of Training-Free Multi-Conditional Generation

Independent Condition Optimization

Most methods optimize each condition separately, assuming independence — this neglects semantic interactions, leading to incoherent or conflicting outputs [6]

Conflicting Guidance Signals

Without a learned fusion mechanism, different conditions may pull the generation in different directions, degrading visual fidelity [8]

Lack of Joint Condition Satisfaction

Existing methods struggle to enforce joint constraint satisfaction, especially when conditions interact non-trivially [21]

Objectives

1

- Create a flexible base solution for training-free multi conditional image generation
- Base solution should be able to handle multiple conditions and remain training-free even when new conditions are added
- Base solution should be flexible and open to enhancements (for example in Objective 2)

2

- Solve the problem of the lack of handling of complicated and non-linear interactions between multiple potentially dependent conditions

Literature Review

Outline

Conditional Score Based Diffusion Models

Energy Diffusion Guidance

Approximating Time-Dependent Energy Guidance with Time-Independent Distance Functions

Conditional Score Based Diffusion Models

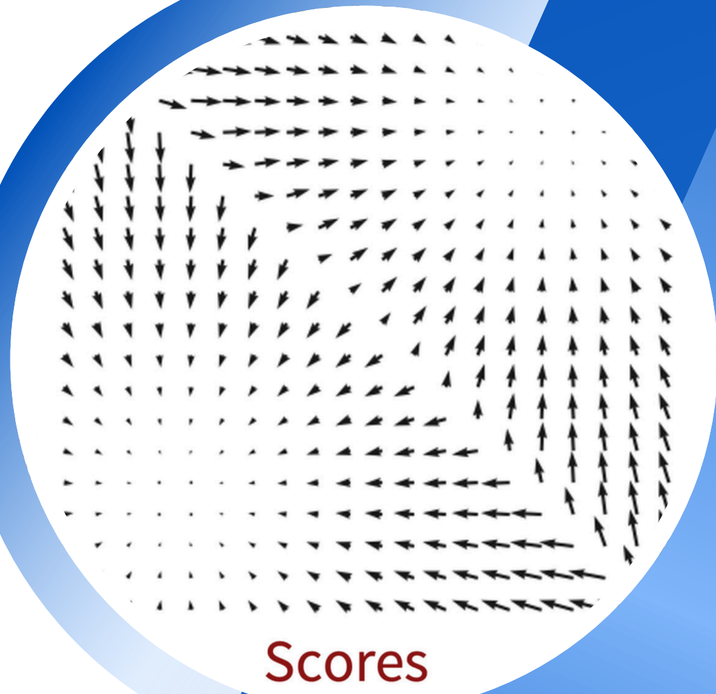
Unconditional Score-Based Diffusion Models (USBDM)

- Operates on score theory
- Learn and estimate a time-dependent score function that guides the denoising phase of a noisy image x_t to x_{t-1} at time step t during the iterative denoising process [16]

Denoising Formula

$$x_{t-1} = \left(1 + \frac{1}{2}\beta_t\right)x_t + \beta_t \nabla_{x_t} \log p(x_t) + \sqrt{\beta_t}\epsilon$$

Score Function



Conditional Score Based Diffusion Models

Conditional Score-Based Diffusion Models (CSBDM)

- A corrective gradient is added to the USBDM denoising formula
- The corrective gradient guides x_t to a hyperplane in the data space that aligns with the condition c [16]

Denoising Formula

$$x_{t-1} = \left(1 + \frac{1}{2}\beta_t\right) x_t + \beta_t \nabla_{x_t} \log p(x_t) + \underline{\nabla_{x_t} \log p(c|x_t)} + \sqrt{\beta_t}\epsilon$$

Corrective Gradient

Conditional Score Based Diffusion Models

Training-Required VS Training-Free Methods

To obtain the conditional corrective gradient $\nabla_{x_t} \log p(c|x_t)$

Training-Required

- Often retain the time-dependent nature of the conditional corrective gradient
- Learn it through approaches like classifier training
- With each new condition, retraining will be required to learn a new corrective gradient
- Conditional information is integrated during the training phase

Training-Free Methods

- In contrast, often aim to approximate the corrective gradient using time-independent functions
- With each new condition, retraining is not required
- Conditional information is incorporated only during inference (denoising)

Energy Diffusion Guidance

Definition of Energy Diffusion Guidance

$$p(c|x_t) = \frac{e^{-\lambda \mathcal{E}(x_t, c)}}{z} \quad [9]$$

Where

1. $z = \int_{c \in \mathcal{C}} e^{-\lambda \ell(x_t, c)}$ represents the normalizing constant,
2. λ represents the temperature
3. $\mathcal{E}(x_t, c)$ represents the energy function measuring the similarity between a given condition c and a noisy image x_t

Energy Diffusion Guidance

Alternative method to model the conditional corrective gradient

$$p(c|x_t) = \frac{e^{-\lambda \mathcal{E}(x_t, c)}}{z}$$

- Energy function value decreases as the similarity between x_t and c increases
- Energy function = zero when x_t and c are perfectly similar
- Thus, the corrective gradient can be remodelled to energy guidance as

$$\nabla_{x_t} \log p(c \mid x_t) \propto -\nabla_{x_t} \varepsilon(x_t, c) \qquad [9]$$

Final denoising formula with energy diffusion guidance

$$x_{t-1} = \left(1 + \frac{1}{2}\beta_t\right) x_t + \beta_t \nabla_{x_t} \log p(x_t) + \sqrt{\beta_t}\epsilon - p_t \nabla_{x_t} \mathcal{E}(x_t, c)$$

Approximating Time-Dependent Energy Guidance with Time-Independent Distance Functions

Training-Required Methods

- Most train classifiers to calculate a Time-Dependent Distance Function between x_t and c

  $D_\phi(c, x_t, t)$

  where ϕ represents the trained parameters of the classifier [5]
- Time-dependent Distance Function then used to approximate the energy guidance function

  $\mathcal{E}(x_t, c) \approx D_\phi(c, x_t, t)$

Training-Free Methods

- Approximate the Time-Dependent Distance Function with Time-Independent Distance Functions

  $D_\phi(C, x_t, t) \approx D_\theta(C, x_0)$

  where θ represents the parameters of a pre-trained classifier [20]
- Time-Independent Distance Function then used to approximate the energy guidance function

  $\mathcal{E}(x_t, c) \approx D_\theta(C, x_0)$

Solutions

Outline

- MATCHE Diffusion
- Multi-Conditional Time-Independent Approximated Energy Guidance
- Approximating Clean Image from Noisy Image
- Interaction Modelling

Solutions Overview

Unanswered Problems?

1. How do we use energy-based guidance to build a flexible base mutli-conditional training-free denoising process?

2. If we were to use the time-independent approximation of energy-based guidance for the training-free denosing process, how do we get the required clean image x_0 from noisy image x_t at each time step t ?

$$D_\phi(C, x_t, t) \approx D_\theta(C, x_0)$$

3. How do we solve the common problem of training-free image generation approaches - lack of handling complicated interactions between multiple potentially dependent conditions?

Solutions Overview

MATCHE DIFFUSION

Multi-conditional
Approximated
Time-independent
Condition-**H**armonizing
Energy guidance

MATCHE DIFFUSION

Multi-conditional **A**pproximated **T**ime-independent
Condition-**H**armonizing **E**nergy guidance

Main components

Multi-Conditional Approximated Time-Independent Energy Guidance

- Base denoising process supporting multiple conditions, while remaining training free
- Flexible to enhancements

Clean Image Approximation

- Approximation of clean image $x_{0|t}$ from noisy image x_t

Interaction Modelling (Condition Harmonizing)

- Captures complex, non-linear interdependencies between conditions

Multi-Conditional Time-Independent Approximated Energy Guidance

No Interaction Modelling

Definition

$$\mathcal{E}(x_t|c_1, c_2, \ldots, c_n) \approx \sum_{i=1}^{n} \lambda_i D_i(c_i, x_{0|t})$$

Where

- c_i represents a given condition i
- $x_{0|t}$ represents the approximated clean image at time step t
- $D_i(c_i, x_{0|t})$ represents the distance between condition i and $x_{0|t}$ computed by a pre-trained network that is specific to condition i
- λi represents the weighting factor of $D_i(c_i, x_{0|t})$

Example

If c_i represents a text condition, then the pre-trained network i could be a CLIP embedding model, and $D_i()$ could be a Euclidean distance value between the CLIP embedding of c_i and $x_{0|t}$

Multi-Conditional Time-Independent Approximated Energy Guidance

With Interaction Modelling

Definition

$$\mathcal{E}(x_t|c_1, c_2, \ldots, c_n) \approx \sum_{i=1}^{n} \lambda_i D_i(c_i, x_{0|t}) + \sum_{i \neq j}^{n} \lambda_{ij} \Phi_{ij}(x_{0|t}, c_i, c_j)$$

Where

- $\Phi_{ij}(x_{0|t}, c_i, c_j)$ represents a function that models the interactions between conditions c_i , c_j and the approximated clean image $x_{0|t}$ in their respective spaces
- λij represents the weighting factor of $\Phi_{ij}(x_{0|t}, c_i, c_j)$

Approximating Clean Image $x_{0|t}$ from Noisy Image x_t

$$x_{0|t} \approx E[x_0|x_t] = \frac{1}{\sqrt{\bar{a}_t}}(x_t + (1-\bar{a}_t)s(x_t,t))$$

1. **Forward Process (Gaussian Noise Addition)**

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0,\mathbf{I}), \quad \alpha_t = \prod_{i=1}^{t}(1-\beta_i)$$

2. **Rearranging to Estimate X0**

$$\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}\right)$$

3. **Since the score-based diffusion model, s(, t) learns to predict ϵ, or its scaled version**

$$\boldsymbol{\epsilon} \approx -\sqrt{1-\bar{\alpha}_t}\cdot s(\mathbf{x}_t,t)$$

4. **Sub (3) in (2)**

$$\mathbf{x}_0 \approx \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1-\bar{\alpha}_t)\,s(\mathbf{x}_t,t))$$

Interaction Modelling

Modelling $\Phi_{ij}(x_{0|t}, c_i, c_j)$ in $\mathcal{E}(x_t|c_1, c_2, \ldots, c_n) \approx \sum_{i=1}^{n} \lambda_i D_i(c_i, x_{0|t}) + \sum_{i \neq j}^{n} \lambda_{ij} \Phi_{ij}(x_{0|t}, c_i, c_j)$

Euclidean Distance

$$\Phi_{ij}(x_{0|t}, c_i, c_j) = \|c_i - c_j\|$$

Cosine Similarity

$$\Phi_{ij}(x_{0|t}, c_i, c_j) = \frac{c_i \cdot c_j}{\|c_i\| \|c_j\|}$$

Pearson Correlation

$$\Phi_{ij}(x_{0|t}, c_i, c_j) = \frac{\text{cov}(c_i, c_j)}{\sigma(c_i) \cdot \sigma(c_j)}$$

Polynomial Kernel

$$\Phi_{ij}(x_{0|t}, c_i, c_j) = (D_i(c_i, x_0|t) \cdot D_j(c_j, x_0|t) + k)^p$$

Sigmoid Kernel

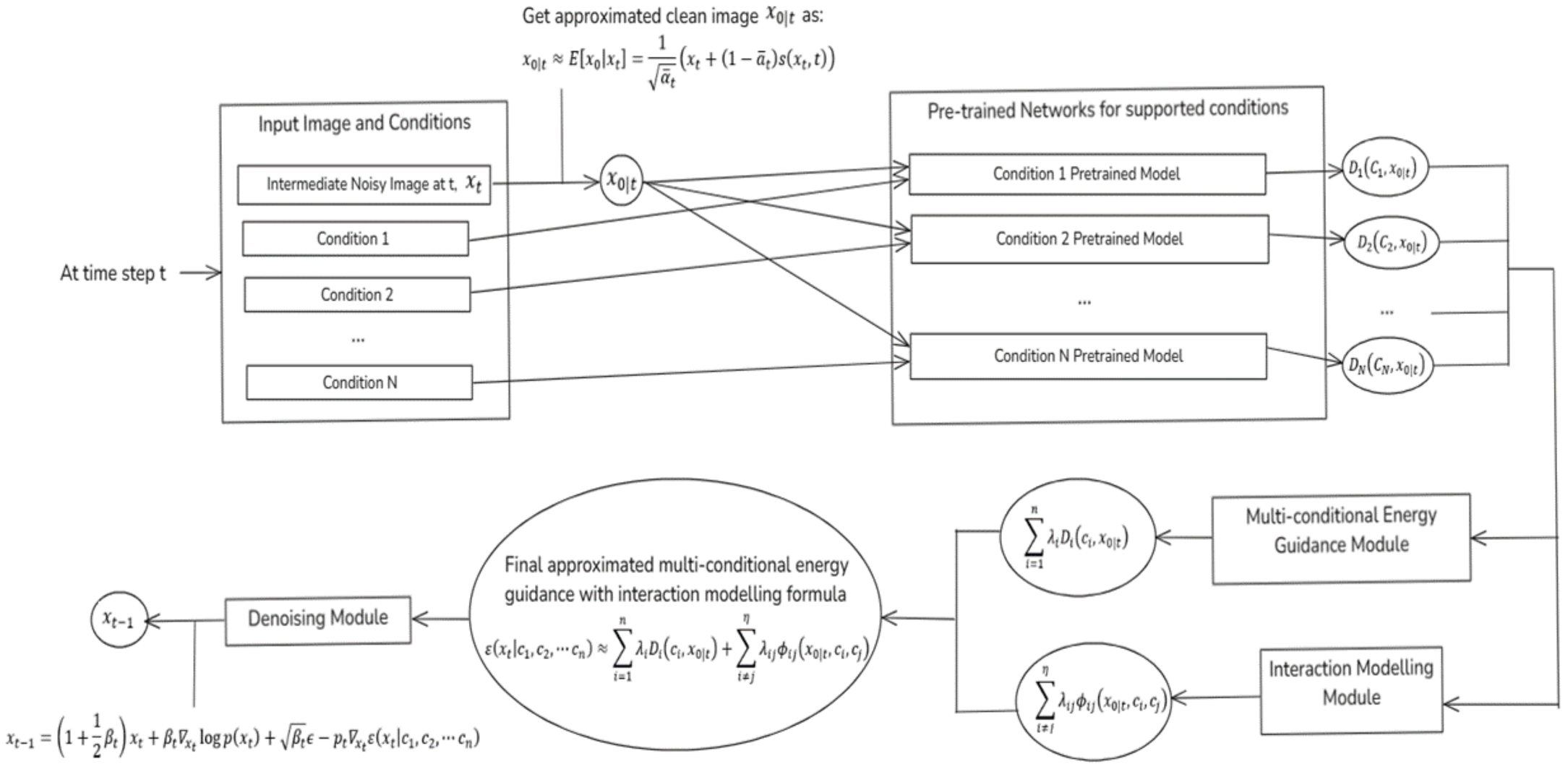
$$\Phi_{ij}(x_{0|t}, c_i, c_j) = tanh(\alpha \cdot D_i(c_i, x_0|t) \cdot D_j(c_j, x_0|t) + m)$$

Gaussian Kernel

$$\Phi_{ij}(x_{0|t}, c_i, c_j) = G(c_i, x_{0|t}) \cdot G(c_j, x_{0|t}) \cdot G(c_i, c_j)$$

where $G(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$

MATCHE Architecture



Experimental Setup

Outline

Scope

Configuration

Experiment Setup

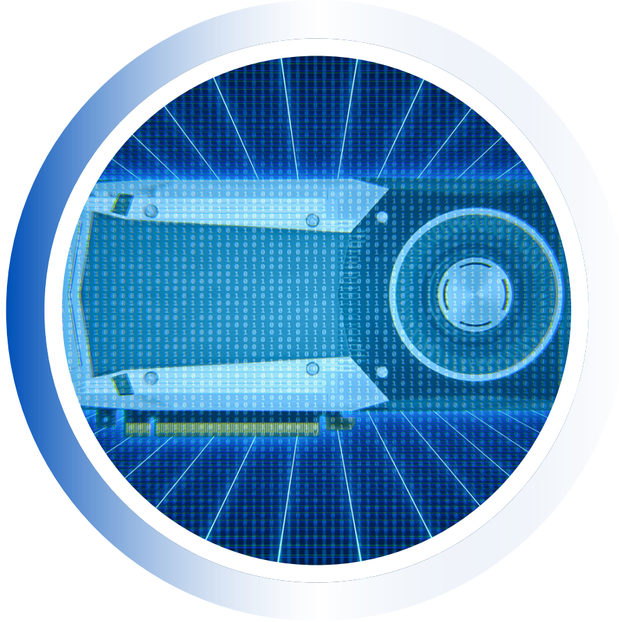
Hyperparameter Optimisation

Scope

Image Generation Model

Limitations

- Given GPU memory constraints, running larger diffusion models with general image generation capabilities (such as Stable Diffusion or ControlNet) was infeasible
- These models require substantial Video Random-Access Memory (VRAM) to store high-dimensional tensors during the iterative denoising process, leading to out-of-memory errors on my available hardware
- My NVIDIA GeForce GTX 1660 Ti has 6GB of VRAM, Stable Diffusion and ControlNet requires 8-12GB of VRAM

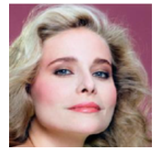
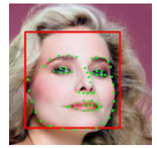


Chosen Model

- Pre-trained unconditional human face diffusion model

Scope

Conditions (Specific to facial image generation)

Condition	Description	Illustration
Face ID	Encodes identity-specific facial features in a numerical representation, ensuring consistency in facial appearance	
Sketch	Provides a structural outline of the subject, capturing overall shape and contours.	
Landmark	Defines spatial key points (e.g., eyes, nose, mouth) to enforce geometric accuracy in facial features	
Segmentation Map	Specifies region-based attributes, guiding the model in differentiating facial parts and background elements	
Text	Textual prompt that offers high-level semantic descriptions, providing flexible and interpretable guidance for image generation	Text Prompt = “Wearing Lipstick”

Configuration

Image Generation Model

- Unconditional Human Face Diffusion Model [25].
  - Supports image resolution of 256×256
  - Pre-trained on human faces in the CelebA-HQ dataset

Configuration

Condition-specific pre-trained model

$$D_i(c_i, x_{0|t}) = Distance_i\left(f_i(c_i), f_i(x_{0|t})\right)$$

Condition, *ci*	Model, *fi(x)*	*Distance i ()*
Face ID	Open-source Human Face Identification Network, ArcFace [4]	Euclidean Distance
Sketch	Open-source Sketch-to-Photo Synthesis Network [18]	Euclidean Distance
Landmark	Open-source Landmark Extractor Network [2]	Euclidean Distance
Segmentation Map	Open-source Real-Time Semantic Segmentation Network BiSeNet [19]	Euclidean Distance
Text	Open-source CLIP Image and Text Encoder [12]	Euclidean Distance

Experimental Setup

Setup Parameters

- Avoid combining conditions with same information
- Generated 1000 images from each multi-condition group
  - (Face ID, Landmark, Text)
  - (Sketch, Landmark, Text)
  - (Face ID, Segmentation Map, Text)
  - (Sketch, Segmentation Map, Text)
- Evaluation criterion
  - Average Condition-Specific Euclidean Distances
  - Average Fréchet Inception Distance (FID)

Hyperparameter Optimisation

Limitations

- Due to GPU limitations which resulted in long image generation times, hyperparameter tuning was conducted using a grid search over a predefined set of values
- This approach ensures a systematic exploration of key parameters while maintaining computational feasibility

Hyperparameter Optimisation

Search Space

$$\mathcal{E}(x_t|c_1, c_2, \ldots, c_n) \approx \sum_{i=1}^{n} \lambda_i D_i(c_i, x_{0|t}) + \sum_{i \neq j}^{n} \lambda_{ij} \Phi_{ij}(x_{0|t}, c_i, c_j)$$

Hyperparameter	Search Space	Description
λi	(By ratio, rij = λi / λij) 1, 10, 100, 1000	Weighting Factor for Distance Terms
λij	(By ratio, ri1j1i2j2 = λi1j1 / λi2j2) 1, 10, 100, 1000	Weighting Factor for Interaction Terms
p	1, 2, 3, 4, 5	Polynomial Degree for Polynomial Kernel
k	0.5, 1, 5, 10, 15	Constant for Polynomial Kernel
α	0.05, 0.1, 0.5, 1, 2	Scaling factor for Sigmoid Kernel
m	0.5, 1, 5, 10, 15	Constant for Sigmoid Kernel
σ	0.5, 0.8, 1	Standard deviation for Gaussian Kernel

Results

Outline

- Hyperparameters
- Quantitative Results
- Qualitative Results

Hyperparameters

Grid Search Results

Hyperparameter	Value	Description
λi	$\lambda\ \{Text\} : \lambda\ \{Seg\ Map\} : \lambda\ \{Landmark\} : \lambda\{ID\} : \lambda\ \{Sketch\}$ = 1000 : 1 : 1000 : 1000 : 10	Weighting Factor for Distance Terms
λij	For all i, j, λij = 1	Weighting Factor for Interaction Terms
p	3	Polynomial Degree for Polynomial Kernel
k	1	Constant for Polynomial Kernel
α	1	Scaling factor for Sigmoid Kernel
m	1	Constant for Sigmoid Kernel
σ	0.5	Standard deviation for Gaussian Kernel

Quantitative Results

Comparison between Interaction Models

	Text		Facial ID		Segmentation Map		Landmark		Sketch	
Interaction Models	**FID**	**Distance**	**FID**	**Distance**	**FID**	**Distance**	**FID**	**Distance**	**FID**	**Distance**
MATCHE (No Interaction Modelling)	113	21.8	158	84.1	129	2331	139	17.3	119	329
Euclidean Distance	163	34.9	188	132	146	2737	136	18.9	129	368
Cosine Similarity	130	24.6	144	73.5	118	2147	128	21.4	116	317
Pearson Correlation	149	29.2	148	83.2	131	2722	141	16.1	131	396
Polynomial	86.2	18.0	102	71.5	91.0	2174	95.3	15.3	91.4	293
Sigmoid	102	21.7	118	68.4	101	2272	127	16.3	122	322
Gaussian	74.2	13.4	86.3	58.7	77.5	1825	81.7	14.9	89.2	248

Quantitative Results

TediGAN Baseline Comparison

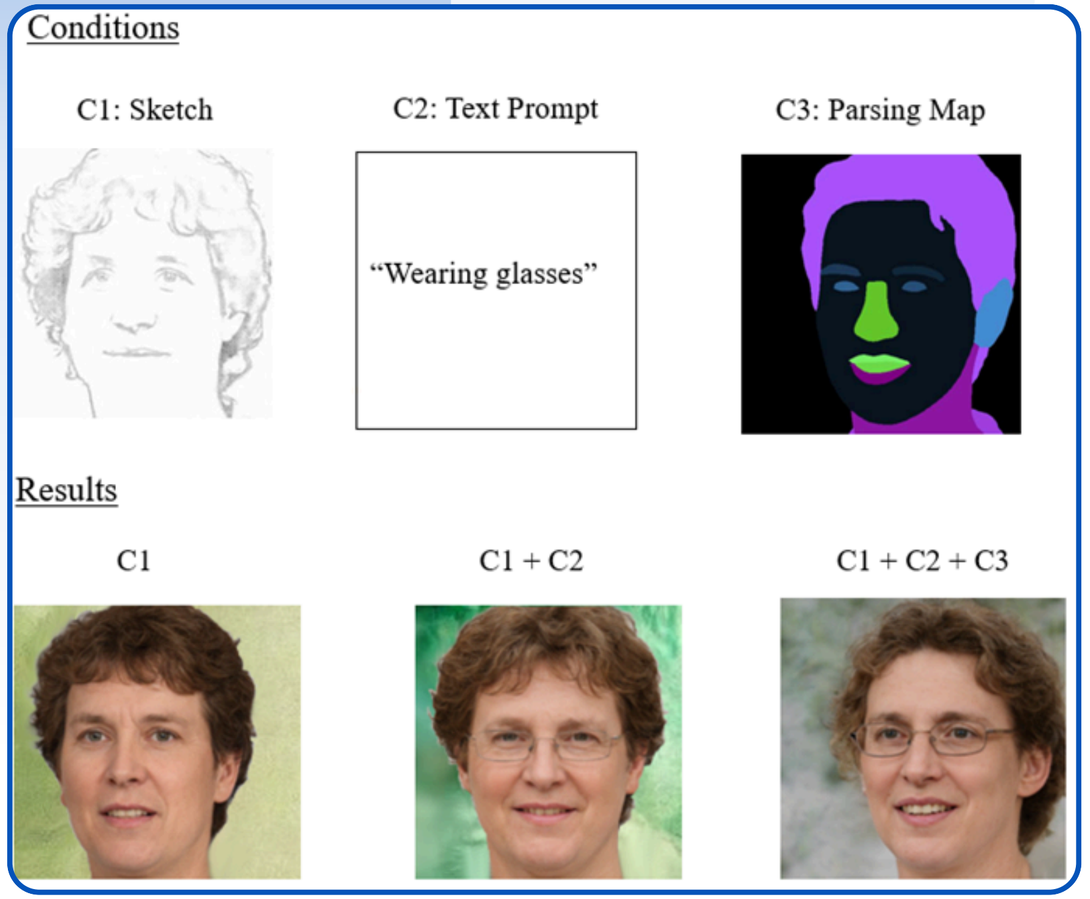
- To further validate the effectiveness of MATCHE diffusion, we compared MATCHE diffusion with Gaussian Kernels, with the baseline model TediGAN.
- TediGAN is a multi-modal image generation framework that enables text-guided synthesis through StyleGAN inversion and visual-linguistic similarity learning [17]

	Text		Facial ID		Segmentation Map		Landmark		Sketch	
Interaction Models	**FID**	**Distance**	**FID**	**Distance**	**FID**	**Distance**	**FID**	**Distance**	**FID**	**Distance**
TediGAN [17]	71.9	12.8	82.8	59.0	81.2	2073	87.0	19.4	89.2	248
MATCHE	74.2	13.4	86.3	58.7	77.5	1825	79.7	14.9	98.2	288

Qualitative Results

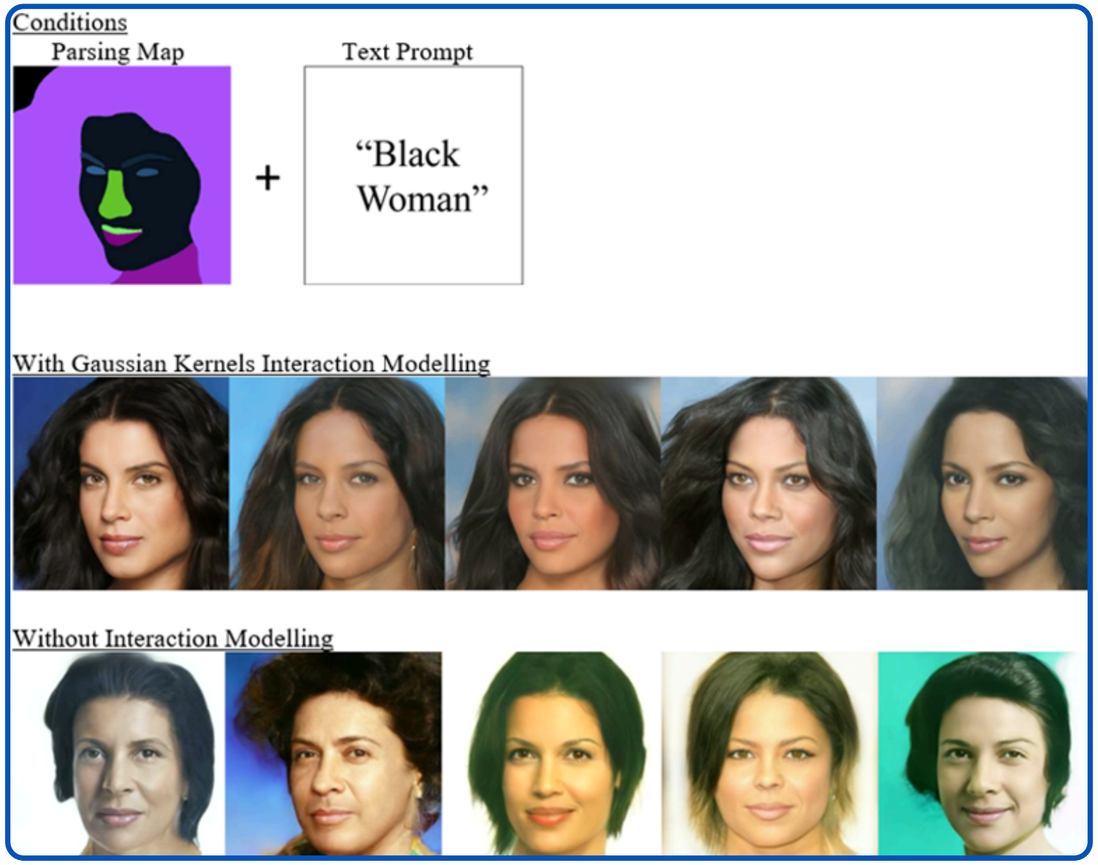
Illustration of Sequential Multi-Conditional Image Generation MATCHE Diffusion

- Shows a sequence of image generation to better illustrates our model's ability to integrate multiple conditions progressively
- In this example:
  - (Sketch → Sketch + Text Prompt → Sketch + Text Prompt + Segmentation Map)
- This demonstrates how our proposed framework is able to add additional constraints and refine the output while preserving visual coherence



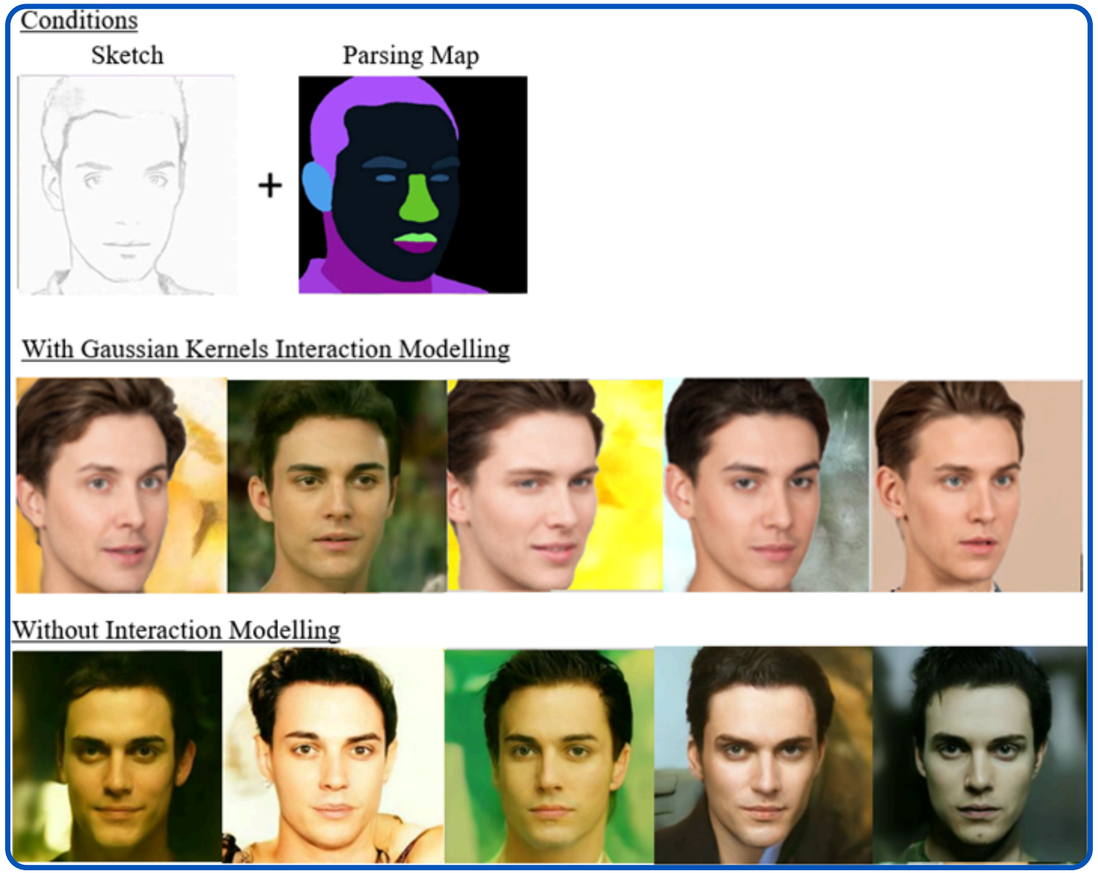
Qualitative Results

Segmentation Map + Text Prompt, Multi-Conditional Image Generation Result



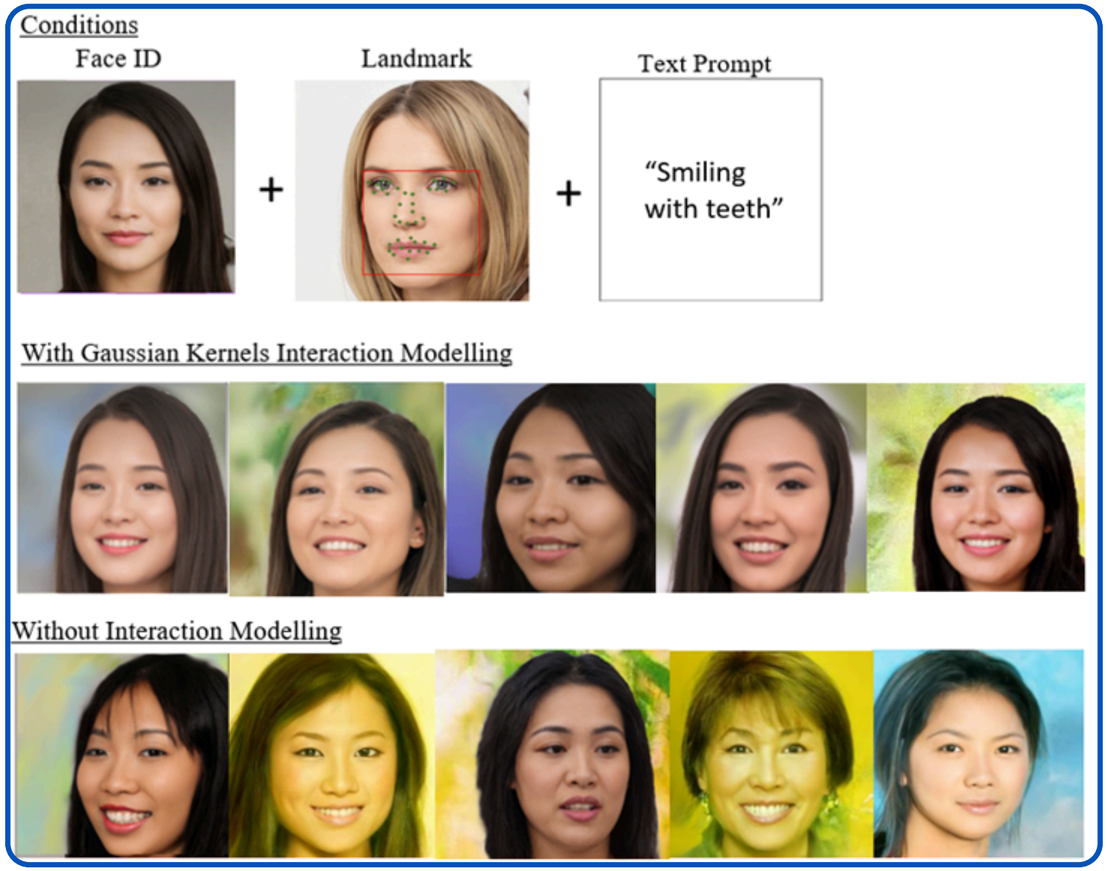
Qualitative Results

Sketch + Segmentation Map, Multi-Conditional Image Generation Result



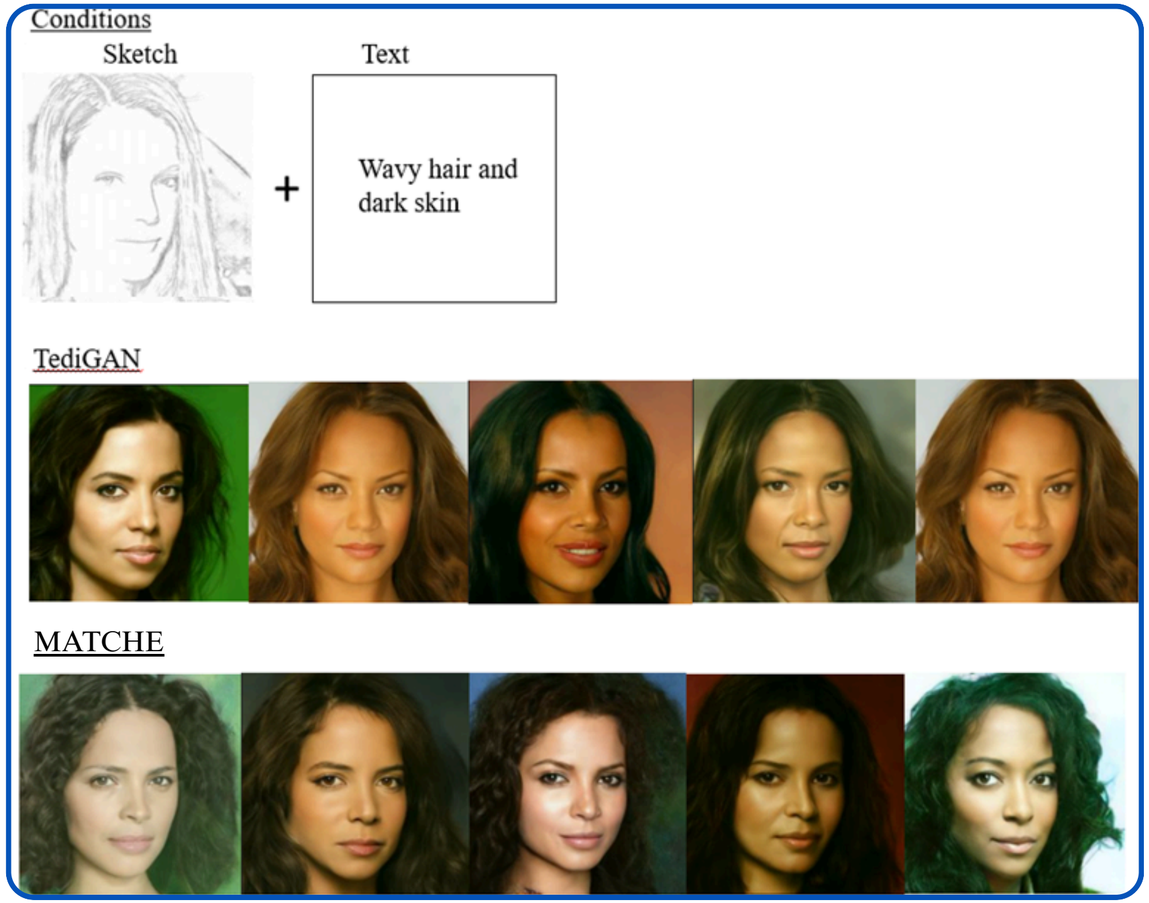
Qualitative Results

Face ID + Landmark + Text Prompt, Multi-Conditional Image Generation Result



Qualitative Results

Baseline Comparison: Sketch + Text Prompt



Evaluation

Outline

Results Evaluation

Experimental Limitations

Future Work

Results Evaluation

Interation Modelling

- Interaction modelling greatly improves quality of images
  - Seen from lower FID and improved visual fidelity
- Appropriate interaction modelling is crucial, inappropriate modelling may result in worse results than no interaction modelling at all
  - Simple euclidean distance and cosine similarity generated worse results than base model with no interaction modelling

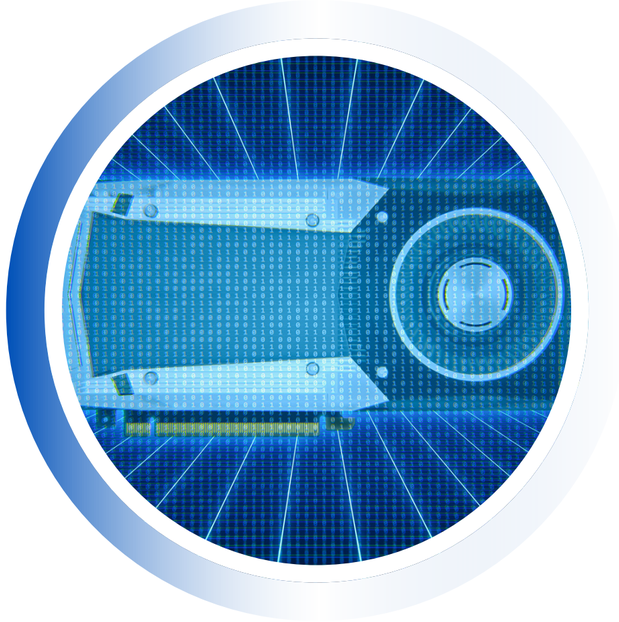
Baseline Comparison

- Proposed method remains competitive with state-of-the-art networks like TediGAN
- Highlights the incorporating MATCHE into existing training-free multi conditional methods to improve quality of outputs

Experimental Limitations

Model and Condition Scope

- Given GPU memory constraints, running larger diffusion models with general image generation capabilities (such as Stable Diffusion or ControlNet) was infeasible
- My NVIDIA GeForce GTX 1660 Ti has 6GB of VRAM, Stable Diffusion and ControlNet requires 8-12GB of VRAM
- Thus, the model chosen was a smaller pre-trained unconditional face generation model
- Consequently, the conditions tested were specific to facial images
- However, the proposed time-independent training-free multi-conditional energy guidance framework is model and condition agnostic and thus the results remain translatatble to general image generation



Experimental Limitations

Hyperparameter Optimisation

- Image generation process was highly time-intensive due to GPU resource constraints
- Particularly when exploring multiple hyperparameter combinations.
- Extremely high number of possible configurations for interaction model-specific hyperparameters, along with the weighting factors for each condition, was extremely large.
- As a result, we acknowledge that our proposed grid search optimization method and the defined search space may not have been the most efficient approach for hyperparameter tuning.
- Consequently, the results for the optimised parameters presented may not represent the most optimal settings.

Future Work

Anisotropic Gaussian Kernels

Standard Gaussian Kernel

$$K(x, y) = e^{-\sum_{i=1}^{d} \frac{(x_i - y_i)^2}{2\sigma_i^2}}$$

- σi is a fixed value, assumed to be fixed across all dimensions [13]
- This is to reduce the hyperparameter search space to address GPU limitations and minimize computational complexity.
- However, with the fixed σi, the standard Gaussian Kernel assumes equal influence in all directions, but some conditions (e.g., shape vs. color) influence different aspects of the image

Anisotropic Gaussian Kernel

$$K(x, y) = e^{-\frac{(x-y)^T \Sigma^{-1} (x-y)}{2}}$$

- Σ is a predefined diagonal covariance matrix with varying σi values based on the perpetual importance of the different dimensions [1]
- This flexibility enables the kernel to capture the anisotropic nature of images, where different features (dimensions) may have different importance or units, requiring different levels of smoothness or variance along each axis [1]
- Can potentially add improvements to shape, spatial, and color constraints by ensuring conditions affect only relevant aspects of image features

Future Work

Extensions to 3D Reconstruction

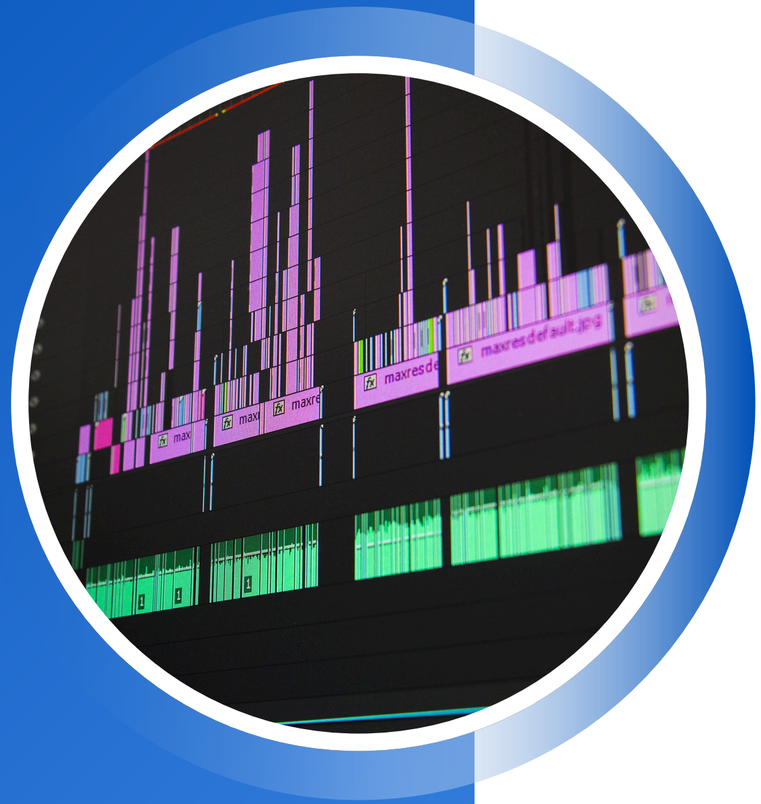
- 3D reconstruction can be viewed as generating a 3D representation X that best matches multiple observations or constraints.
- With any 3D diffusion model, we can incorporate MATCHE into the denoising process as well
- (e.g., voxel grid - 3D CNNs, GANs; point cloud - PointNet++)

MATCHE

$$\mathcal{E}(x_t|c_1, c_2, \ldots, c_n) \approx \sum_{i=1}^{n} \lambda_i D_i(c_i, x_{0|t}) + \sum_{i \neq j}^{n} \lambda_{ij} \Phi_{ij}(x_{0|t}, c_i, c_j)$$

Where

- X: the reconstructed 3D object (e.g., voxel grid, mesh, point cloud)
- ci: different input conditions (e.g., RGB image, silhouette, multi-view images, depth, text)
- Di: discrepancy function for each condition (e.g., perceptual distance between projected image and original)
- Φij: interaction terms (e.g., consistency between silhouette and text, multi-view alignment)



Future Work

Extensions to Video Synthesis

Improved Spatial Energy Guidance

$$\mathcal{E}^k_{\text{spatial}} = \mathcal{E}(x_t \mid c^k_1, c^k_2, \ldots, c^k_n)$$
$$\approx \sum_{i=1}^{n} \lambda_i D_i(c^k_i, x_{0|t} t^k) + \sum_{i \neq j}^{n} \lambda_{ij} \Phi_{ij}(x^k_{0|t}, c^k_i, c^k_j)$$

- MATCHE can be adapted to energy guidance frame-by-frame in video synthesis
- Enhance temporal smoothness by integrating interaction modelling
- Modelling these interactions well can ensure smooth transitions, maintain motion coherence, and dynamic content effectively
- Enable the generator to produce sequences where each frame is consistent with its neighbors, preserving the continuity and flow necessary for realistic video content

Future Work

Extensions to Video Synthesis

Temporal Energy Guidance (e.g. Optical flow warping loss)

$$\begin{aligned}\mathcal{E}^{(k)}_{\text{temporal}} &= \beta \cdot \mathcal{T}(X_0^{(k-1)}, X_0^{(k)}) \\ &= \beta \cdot \left\| X_0^{(k)} - \text{warp}(X_0^{(k-1)}) \right\|_2^2\end{aligned}$$

↓

Total Energy Guidance over the Video

$$\mathcal{E}_{\text{total}} = \sum_{k=1}^{T} \mathcal{E}^{k}_{\text{spatial}} + \mathcal{E}^{k}_{\text{temporal}}$$

↓

Final Denoising Formula

$$X_{t-1}^{(k)} = X_t^{(k)} - \Delta^{(k)}_{\text{diffusion}} - \eta \cdot \nabla_{X_t^{(k)}} \mathcal{E}_{\text{total}} \quad [22]$$

where $\Delta^{(k)}_{\text{diffusion}}$ is the base model's denoising step (e.g., from DDPM or DDIM).

Conclusion

Proposed Solution

MATCHE
Multi-conditional **A**pproximated **T**ime-independent **C**ondition-**H**armonizing **E**nergy guidance

Demonstrated Effectiveness

Outperformed the TediGAN baseline in some conditions with notable improvements in both qualitative and quantitative metrics.

Future Directions

- Explore alternative kernel methods for better scaling
- Extensions to other generative tasks
  - 3D Reconstruction
  - Video Synthesis

Addressed Key Limitations

- Supports multiple conditions
- Training-free
- Condition harmonization - interaction modelling

Acknowledged Limitations

- GPU constraints impacting hyperparameter tuning
- Limits the experimental scope

This project lays a solid foundation for advancements in training-free multi-conditional image generation, opening avenues for more robust generative applications such as video synthesis and 3D reconstruction.

References

References

[1] Berry, T., and Sauer, T., 'Local kernels and the geometric structure of data', arXiv, https://doi.org/10.48550/arXiv.1407.1426, 2014.

[2] Chen, C., 'PyTorch Face Landmark: A fast and accurate facial landmark detector', github, https://github.com/cunjian/pytorch_face_landmark, 2021.

[3] Chung, H., Sim, B., and Ye, J. C., 'Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction', arXiv, https://doi.org/10.48550/arXiv.2112.05146, 2022.

[4] Deng, J., Guo, J., Xue, N., and Zafeiriou, S., 'Arcface: Additive angular margin loss for deep face recognition', arXiv, https://doi.org/10.48550/arXiv.1801.07698, 2019.

[5] Dhariwal, P., and Nichol, A., 'Diffusion models beat gans on image synthesis', arXiv, https://doi.org/10.48550/arXiv.2105.05233, 2021.

[6] Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D., 'Prompt-to-Prompt Image Editing with Cross Attention Control', arXiv, https://doi.org/10.48550/arXiv.2208.01626, 2022

[7] Hofmann, T., Schölkopf, B., and Smola, A. J., 'Kernel methods in machine learning', arXiv, https://doi.org/10.48550/arXiv.math/0701907, 2008.

[8] Kim, G., Kwon, T., and Ye, J., 'DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation', arXiv, https://doi.org/10.48550/arXiv.2110.02711, 2021

[9]LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F., 'A tutorial on energy-based learning', MIT Press, http://yann.lecun.com, 2006.

[10] Lin, H.-T., and Lin, C.-J., 'A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods', 2003

[11] Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S., 'SDEdit: Guided image synthesis and editing with stochastic differential equations', arXiv, https://doi.org/10.48550/arXiv.2108.01073, 2022.

References

[12] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 'Learning transferable visual models from natural language supervision', arXiv, https://doi.org/10.48550/arXiv.2103.00020, 2021.

[13] Rasmussen, C. E., and Williams, C. K. I., 'Gaussian processes for machine learning', MIT Press, https://doi.org/10.7551/mitpress/3206.001.0001, 2006.

[14] Schölkopf, B., & Smola, A. J., 'Learning with kernels: Support Vector Machines, regularization, optimization, and beyond.', MIT Press, https://doi.org/10.7551/mitpress/4175.001.0001, 2001

[15] Shawe-Taylor, J., and Cristianini, N., 'Kernel methods for pattern analysis', Cambridge University Press, https://doi.org/10.1017/CBO9780511809682, 2004.

[16] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B., 'Score-based generative modelling through stochastic differential equations', arXiv, https://doi.org/10.48550/arXiv.2011.13456, 2021.

[17] Xia, W., Yang, Y., Xue, J.-H., and Wu, B., 'TediGAN: Text-guided diverse face image generation and manipulation', arXiv, https://doi.org/10.48550/arXiv.2012.03308, 2021.

[18] Xiang, X., Liu, D., Yang, X., Zhu, Y., Shen, X., and Allebach, J. P, 'Adversarial open domain adaptation for sketch-to-photo synthesis', arXiv, https://doi.org/10.48550/arXiv.2104.05703, 2022.

[19] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., and Sang, N., 'Bisenet: Bilateral segmentation network for real-time semantic segmentation', arXiv, https://doi.org/10.48550/arXiv.1808.00897, 2018.

[20] Yu, J., Wang, Y., Zhao, C., Ghanem, B., and Zhang, J., 'FreeDoM: Training-free energy-guided conditional diffusion model', arXiv, https://doi.org/10.48550/arXiv.2303.09833, 2023.

[21] Zhang, L., Rao, A., Agrawala, M., 'Adding Conditional Control to Text-to-Image Diffusion Models', arXiv, https://doi.org/10.48550/arXiv.2302.05543, 2023.

[22] Zhang, Y., Zhao, D., Luo, Z., Chen, D., Huang, Y., Wang, L., Shen, Y., Zhou, J., Tan, T., 'VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation', arXiv, https://doi.org/10.48550/arXiv.2302.05543, 2023.

Thank You!

Special thanks to my project supervisor, Professor Lu Shijian, and my student mentor, Xu Muyu, for their invaluable guidance and support throughout this journey!

Q&A

Appendix

Appendix 1

Derivation of Approximating Clean Image X{0 | t} from Noisy Image X{t}

$$x_{0|t} \approx E[x_0|x_t] = \frac{1}{\sqrt{\bar{a}_t}}(x_t + (1-\bar{a}_t)s(x_t,t))$$

(1) Forward Process (Gaussian Noise Addition)

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0,\mathbf{I})$$

(2) Rearranging to Estimate X0

$$\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}\right)$$

(3) Since the score-based diffusion model learns to predict ϵ, or its scaled version

$$\boldsymbol{\epsilon} \approx -\sqrt{1-\bar{\alpha}_t}\cdot s(\mathbf{x}_t,t)$$

(4) Sub (3) in (2)

$$\mathbf{x}_0 \approx \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1-\bar{\alpha}_t)\,s(\mathbf{x}_t,t))$$

Appendix 2

Experimental Setup - Multi-Conditional Groups Combination

Classified conditions into similar groups based on the type of information they provide

Group	Description	Conditions
Structural Representation	Defines the specifics of the shape and structure of the facial subject	Face ID, Sketch
Spatial Feature	Describes locations of key points and regions	Landmark, Segmentation Map
Semantic Description	Provides high-level conceptual information	Text

Appendix 3

Models

Characteristics \ Technique	Euclidean Distance	Cosine Similarity	Pearson Correlation	Polynomial Kernel	Sigmoid Kernel	Gaussian Kernel
Captures linear relationships	✅	✅	✅	✅	✅	✅
Captures non-linear relationships	❌	❌	❌	✅	✅	✅
Invariant to vector magnitudes	❌	✅	✅	❌	❌	✅
Sensitive to spatial structure	✅	❌	❌	✅	❌	✅
Suitable for high-dimensional data	❌	✅	❌	✅	✅	✅
Captures complex interactions	❌	❌	❌	✅	✅	✅
Computational cost	✅	✅	✅	❌	❌	❌
Robust to noise or scale variance	❌	✅	✅	❌	❌	✅

Appendix 4

Conditional Score Based Diffusion Models

$$x_{t-1} = \left(1 + \frac{1}{2}\beta_t\right) x_t + \beta_t \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(c|x_t) + \sqrt{\beta_t}\epsilon$$

- x_{t-1}
- The **denoised sample at time step** $t-1$.
- This is the updated version of the noisy sample x_t after applying the reverse diffusion step.

- $\left(1 + \frac{1}{2}\beta_t\right) x_t$
- This is the **drift term**, derived from the discretized reverse-time stochastic differential equation (SDE).
- The coefficient $\left(1 + \frac{1}{2}\beta_t\right)$ slightly boosts the previous sample x_t, adjusting it for the variance schedule.
- Think of it as a linear adjustment based on how fast noise is being removed at time t.

- $\beta_t \nabla_{x_t} \log p(x_t)$
- This term is a **prior score term**, where:
  - $\nabla_{x_t} \log p(x_t)$ is the **score function** of the marginal data distribution (how likely is x_t as a data sample).
  - It tells us the direction in which the probability of the sample increases.
- The multiplication with β_t scales the influence of this guidance based on the current noise level.
- Intuition: Push x_t towards regions that look more like real data.

- $\nabla_{x_t} \log p(c|x_t)$
- This is the **conditional score term** (conditioning signal).
- It represents the **gradient of the log-likelihood of the condition** c (e.g., class label, image sketch, identity vector, etc.), given x_t.
- Intuition: Guides the generation process to samples that are not just real (as above), but also **match the condition** c.
- This is how **conditioning** is achieved in training-free conditional generation.

- $\sqrt{\beta_t}\epsilon$
- This is the **stochastic noise term**, where:
  - $\epsilon \sim \mathcal{N}(0, I)$ is Gaussian noise.
  - $\sqrt{\beta_t}$ controls how much noise is injected at time step t.
- It ensures the sampling process remains stochastic and aligned with the original diffusion process.

🧠 **Overall Intuition:**

You're **starting from noise** and denoising gradually over steps $t \to 0$, and in each step:

- You drift slightly in the direction of the current sample.
- You nudge the sample toward high-probability regions of real data $p(x_t)$.
- You further guide it to align with desired conditions $p(c|x_t)$.
- You inject some noise to maintain the stochastic nature of diffusion.

Appendix 5

Diffusion

$$x_{t-1} = \left(1 + \frac{1}{2}\beta_t\right)x_t + \beta_t s(x_t, t) + \sqrt{\beta_t}\,\epsilon - p_t \nabla_{x_t}\mathcal{E}(x_t|c_1, c_2, \ldots, c_n)$$

```
betas = get_beta_schedule(
    beta_schedule="linear",
    beta_start=0.0001,
    beta_end=0.02,
    num_diffusion_timesteps=1000,
)
```

Appendix 5

Diffusion

$$x_{t-1} = \left(1 + \frac{1}{2}\beta_t\right)x_t + \beta_t s(x_t, t) + \sqrt{\beta_t}\,\epsilon - p_t \nabla_{x_t}\mathcal{E}(x_t|c_1, c_2, \ldots, c_n)$$

```
betas = get_beta_schedule(
    beta_schedule="linear",
    beta_start=0.0001,
    beta_end=0.02,
    num_diffusion_timesteps=1000,
)
```

Appendix 5

Diffusion

$$x_{t-1} = \left(1 + \frac{1}{2}\beta_t\right)x_t + \beta_t s(x_t, t) + \sqrt{\beta_t}\,\epsilon - p_t \nabla_{x_t}\mathcal{E}(x_t|c_1, c_2, \ldots, c_n)$$

```
betas = get_beta_schedule(
    beta_schedule="linear",
    beta_start=0.0001,
    beta_end=0.02,
    num_diffusion_timesteps=1000,
)
```