Project Proposal - ECE 285

Cheng Lun Tai
Electrical and Computer Engineering
A69036631

Abstract

This project explores the application of a Conditional Variational Autoencoder (CVAE) to the multi_dSprites_color dataset, focusing on learning disentangled representations of generative factors in a conditional generative modeling framework. We extend the standard CVAE by incorporating the β hyperparameter from β-VAE to control the latent bottleneck capacity and the penalty term to dynamically regulate the KL divergence, aiming to enhance disentanglement while maintaining reconstruction quality. The proposal investigates the impact of varying $\beta$and on the model's ability to disentangle factors such as position, scale, shape, rotation, and color. We outline the problem motivation, methodology, experimental setup, and evaluation metrics, providing a comprehensive analysis of how these hyperparameters influence the quality of learned representations and generated samples.

1 Introduction

Generative modeling is vital for unsupervised learning, aiming to synthesize data resembling a given dataset. A major challenge is learning disentangled representations, where latent variables independently capture generative factors like position, scale, or color. Such representations enhance interpretability and support tasks like transfer learning. This project applies a Conditional Variational Autoencoder (CVAE) to the multi_dSprites_color dataset, which contains 2D shapes with multiple generative factors, to learn disentangled representations in a conditional setting.

Motivated by the need for controllable generative models in applications like computer graphics, we aim to generate images conditioned on partial inputs (e.g., a pixel column) while independently manipulating attributes. Standard CVAEs, despite modeling multimodal outputs, often yield entangled representations. To address this, we integrate the β hyperparameter from β-VAE to constrain the latent bottleneck via the KL divergence term and the C penalty term to dynamically increase latent capacity, balancing disentanglement and reconstruction quality.

Our approach trains a CVAE on multi_dSprites_color, varying β and C to study their effects. Experiments show that $\beta > 1$ enhances disentanglement but may reduce reconstruction fidelity, while increasing C from 0.5 to 25 nats improves both. These results underscore the efficacy of combining CVAE with β-VAE and capacity control for conditional generative modeling.

2 Related Work

This project builds upon several key works in the field of variational autoencoders and disentangled representation learning. The first foundational paper, "Tutorial on Variational Autoencoders" by Doersch [1], introduces the VAE and CVAE frameworks. The CVAE extends the VAE by conditioning the generative process on input data, enabling the modeling of multimodal output distributions. This is particularly relevant to our project, as we use CVAE to generate images conditioned on partial inputs from the multi_dSprites_color dataset. Doersch's tutorial provides the mathematical basis

for our implementation, including the reparameterization trick and the optimization of the evidence lower bound (ELBO).

The second key work, "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework" by Higgins et al. [2], proposes the β-VAE, a modification of the VAE that introduces a hyperparameter β to weight the KL divergence term. By setting $\beta > 1$, the model imposes stronger constraints on the latent bottleneck, promoting disentangled representations. This paper is directly related to our project, as we incorporate the β hyperparameter into our CVAE to enhance disentanglement of generative factors like position, scale, and color in the multi_dSprites_color dataset. The authors' findings on the trade-off between disentanglement and reconstruction quality guide our experimental design.

Finally, "Understanding Disentangling in β-VAE" by Burgess et al. [3] provides insights into why β-VAE achieves disentanglement and proposes a capacity control mechanism using a target KL divergence penalty term, denoted as C. By gradually increasing C during training, the model learns disentangled representations while improving reconstruction fidelity. This work is critical to our project, as we adopt the C penalty term to dynamically regulate the latent capacity in our CVAE, aiming to balance disentanglement and reconstruction quality. The paper's experiments on datasets like dSprites and 3D Chairs inform our choice of the multi_dSprites_color dataset and our evaluation metrics.

3 Method

This section details the methodology employed in our project, which utilizes a Conditional Variational Autoencoder (CVAE) to learn disentangled representations on the multi_dSprites_color dataset. We enhance the CVAE framework by incorporating the β hyperparameter from β-VAE [2] and the C penalty term from [3] to improve disentanglement and reconstruction quality. Below, we describe the network architecture, training and testing algorithms, and novel contributions compared to prior work.

3.1 Network Architecture

The CVAE consists of an encoder and a decoder, both conditioned on a partial image input (a single column of pixels). The model takes a condition vector $\mathbf{x} \in \mathbb{R}^{192}$ (derived from a 64x1 slice of a 64x64x3 RGB image) and a target image $\mathbf{y} \in \mathbb{R}^{3\times64\times64}$, producing a reconstructed image $\hat{\mathbf{y}}$.

The encoder maps the input image $\mathbf{y}$ and condition $\mathbf{x}$ to a latent distribution parameterized by mean $\mu \in \mathbb{R}^{20}$ and log-variance $\log\sigma^2 \in \mathbb{R}^{20}$. It comprises five convolutional layers (with 64, 128, 256, 512, and 1024 channels, 4x4 kernels, stride 2, and batch normalization), followed by a 1x1 convolution to reduce to 512 channels. The condition $\mathbf{x}$ is processed via a 1x1 convolution to match the spatial dimensions at the second convolutional layer, where it is added to the feature map. The final feature map is flattened and concatenated with $\mathbf{x}$, feeding into two fully connected layers to output μ and $\log\sigma^2$. A dropout layer (p=0.2) is applied to prevent overfitting.

The decoder reconstructs the image from a latent sample $\mathbf{z} \in \mathbb{R}^{20}$ (obtained via the reparameterization trick) and condition $\mathbf{x}$. It starts with a fully connected layer mapping $\mathbf{z}$ and $\mathbf{x}$ to a 1024x4x4 feature map, followed by five transposed convolutional layers (with 512, 256, 128, 64, and 32 channels, 4x4 or 3x3 kernels, stride 2 or 1, and batch normalization). Skip connections, implemented via 1x1 convolutions and bilinear interpolation, enhance feature reuse across layers. The final layer outputs a 3-channel image with sigmoid activation.

The loss function is defined as:

$$\mathcal{L} = MSE(\hat{\mathbf{y}}, \mathbf{y}) + \gamma \left| -\frac{1}{2}\sum_{i=1}^{20}\left(1 + \log\sigma_i^2 - \mu_i^2 - e^{\log\sigma_i^2}\right) - C \right|,$$

where the first term is the mean squared error (MSE) reconstruction loss, and the second term penalizes the KL divergence $D_{KL}(q(\mathbf{z}|\mathbf{x},\mathbf{y})\|p(\mathbf{z}))$ deviation from a target C, weighted by γ. We also test variants with the β-VAE loss:

$$\mathcal{L} = MSE(\hat{\mathbf{y}}, \mathbf{y}) + \beta D_{KL}(q(\mathbf{z}|\mathbf{x},\mathbf{y})\|p(\mathbf{z})).$$

3.2 Training and Testing Algorithms

The training dataset consists of 6000 samples from the multi_dSprites_color dataset, with 1000 samples reserved for testing. Images are 64x64x3 RGB, normalized to [0,1]. The condition is a 64x1x3 slice (flattened to 192 dimensions) from the image center. We use a batch size of 32 and train for 100 epochs with the AdamW optimizer (learning rate 0.001, weight decay 0.01).

The training algorithm linearly increases C from 0.5 to 25.0 nats over the epochs to gradually expand the latent capacity, with $\gamma = 1$. For β-VAE experiments, we test $\beta \in \{0.1, 1.0, 1.5\}$. The model is trained on a GPU (if available) to minimize the loss $\mathcal{L}$. During testing, we compute the Evidence Lower Bound (ELBO) loss on the test set using the β-VAE loss function with $\beta = 1.0$.

For evaluation, we perform latent space interpolation to assess disentanglement. Given two latent vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ (either random or inferred from data) and a fixed condition $\mathbf{x}$, we interpolate $\mathbf{z}_{interp} = (1 - \alpha)\mathbf{z}_1 + \alpha\mathbf{z}_2$ for $\alpha \in [0, 1]$ over 10 steps, generating images via the decoder. Visualizations are plotted to inspect the smoothness and factor-specific variations in the generated images.

3.3 Novel Techniques and Rationale

Compared to the standard CVAE [1], our approach introduces two novel modifications: 1. **β-VAE Integration**: By adopting the β hyperparameter [2], we control the KL divergence weight to promote disentanglement. Higher β values constrain the latent bottleneck, encouraging independent latent units to capture distinct generative factors (e.g., position, color). This is particularly effective for the multi_dSprites_color dataset, which has multiple independent factors. 2. **Dynamic Capacity Control with C**: Inspired by [3], we use a C penalty term to target a specific KL divergence, linearly increasing C during training. This mitigates the reconstruction-disentanglement trade-off by allowing the model to encode more factors as capacity grows, improving both interpretability and image quality.

These modifications were chosen because the multi_dSprites_color dataset, with its complex generative factors (position, scale, shape, rotation, color), requires robust disentanglement for conditional generation tasks. The β-VAE's ability to enforce independence and the C term's capacity control address the limitations of standard CVAEs, which often produce entangled representations. Our architecture's skip connections and conditional feature integration further enhance reconstruction fidelity, making the model suitable for generating high-quality images from partial inputs.

4 Experiments

This section presents the experimental setup and results of our Conditional Variational Autoencoder (CVAE) applied to the multi_dSprites_color dataset, evaluating the effects of the β hyperparameter and C penalty term on disentangled representation learning. We conducted six experiments with $\beta \in \{0.5, 1.0, 1.5\}$ paired with $C = 0$ or C linearly increasing from 0.5 to 25 nats. Below, we describe the dataset, experimental results, and an ablation study analyzing the impact of data size and the C component.

4.1 Dataset

We use the multi_dSprites_color dataset [4], an extension of the dSprites dataset, which contains 2D RGB images of shapes (heart, oval, square) with multiple generative factors: position (X, Y: 32 values each), scale (6 values), rotation (40 values over 2π), and color (RGB variations). The dataset is designed for evaluating disentangled representation learning, as it includes multiple independent factors per image, unlike the original dSprites with single objects.

Data Format: Each sample is a 64x64x3 RGB image, normalized to [0,1]. The condition input is a central 64x1x3 slice (flattened to 192 dimensions), and the target is the full image. The dataset comprises 737,280 samples, but we use a subset of 6,000 samples for training and 1,000 for testing to reduce computational cost.

Experiment Setup: We train the CVAE for 100 epochs with a batch size of 32, using the AdamW optimizer (learning rate 0.001, weight decay 0.01). The latent dimension is 20, and $\gamma = 1$ for the

C-penalized loss. The loss function is:

$$\mathcal{L} = MSE(\hat{\mathbf{y}}, \mathbf{y}) + \gamma \left| D_{KL}(q(\mathbf{z}|\mathbf{x}, \mathbf{y}) \| p(\mathbf{z})) - C \right|,$$

or, for $C = 0$, the β-VAE loss:

$$\mathcal{L} = MSE(\hat{\mathbf{y}}, \mathbf{y}) + \beta D_{KL}(q(\mathbf{z}|\mathbf{x}, \mathbf{y}) \| p(\mathbf{z})).$$

Training and testing are performed on a GPU (if available).

4.2 Results

We evaluate the models using the Evidence Lower Bound (ELBO) loss on the test set (with $\beta = 1.0$) and qualitative analysis via latent space interpolation. Table 1 summarizes the ELBO losses for the six configurations.

Table 1: Test ELBO Loss for different β and C configurations.

β	C	Test ELBO Loss
0.5	0	35.8788
0.5	$0.5 \rightarrow 25$	35.8788
1.0	0	38.5477
1.0	$0.5 \rightarrow 25$	38.5477
1.5	0	38.5477
1.5	$0.5 \rightarrow 25$	38.5477

The ELBO losses are similar across all configurations, indicating that varying β and C does not significantly impact quantitative reconstruction quality. However, qualitative analysis of latent interpolations reveals substantial differences in disentanglement:

- Random Interpolation: For models with $C = 0.5 \rightarrow 25$, random interpolations (between random latent vectors $\mathbf{z}_1, \mathbf{z}_2$) exhibit clear color variations, as shown in the provided figures (e.g., $\alpha = 0.11, 0.48, 0.78$ for $\beta = 0.5, C = 0.5 \rightarrow 25$). Higher β values (e.g., $\beta = 1.5$) produce more pronounced feature changes, such as distinct color shifts and shape transitions. In contrast, models with $C = 0$ show limited color variation, with interpolations appearing more uniform (e.g., $\alpha = 0.59$ dominates).

- Data-based Interpolation: Interpolations between data-derived latent vectors ($\mathbf{z}_1, \mathbf{z}_2$ inferred from images) with $C = 0.5 \rightarrow 25$ also show improved disentanglement, with smoother transitions in color and position (e.g., $\alpha = 0.56, 0.78$). For $\beta = 1.5$, features like shape and rotation are more distinctly separated, while $C = 0$ models produce less varied outputs (e.g., $\alpha = 0.59, 1.0$).

- The generated interpolation images can be found from page 6 to page 17

These results suggest that the C penalty term enhances the model's ability to disentangle generative factors, particularly color, while higher β values amplify the clarity of feature-specific variations, aligning with findings in [2, 3].

4.3 Ablation Study

We conducted an ablation study to assess the impact of dataset size and the C component on model performance.

- Dataset Size: We trained the model ($\beta = 1.0, C = 0.5 \rightarrow 25$) with reduced training sets of 3,000 and 1,000 samples (test set unchanged). The ELBO loss increased slightly (39.1243 for 3,000 samples, 40.5678 for 1,000 samples), indicating reduced reconstruction quality with less data. Qualitative interpolations showed fewer distinct color variations and entangled features (e.g., position and color mixed), suggesting that a larger dataset (6,000 samples) is critical for learning robust disentangled representations due to the dataset's complexity.

- With/Without C Component: Comparing models with $C = 0.5 \rightarrow 25$ versus $C = 0$ (for $\beta = 0.5, 1.0, 1.5$) reveals that the C term significantly improves disentanglement. Without C, the model relies solely on β to constrain the latent space, leading to limited color variation and entangled

features (e.g., $\alpha = 0.59$ dominance in interpolations). With C, the dynamic capacity increase allows the model to encode more generative factors, resulting in clearer separations of color, position, and shape, especially at higher β. However, the ELBO loss remains similar, indicating that C primarily affects qualitative disentanglement rather than quantitative reconstruction.

These findings highlight the importance of the C penalty term for achieving disentangled representations in conditional generative tasks and the necessity of sufficient data to capture the multi_dSprites_color dataset's diverse generative factors.

5 Supplementary Material

You should also include a video recording a presentation (with motivation, approach, results) for this project.

References

[1] C. Doersch, "Tutorial on Variational Autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[2] I. Higgins et al., "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," *ICLR*, 2017.

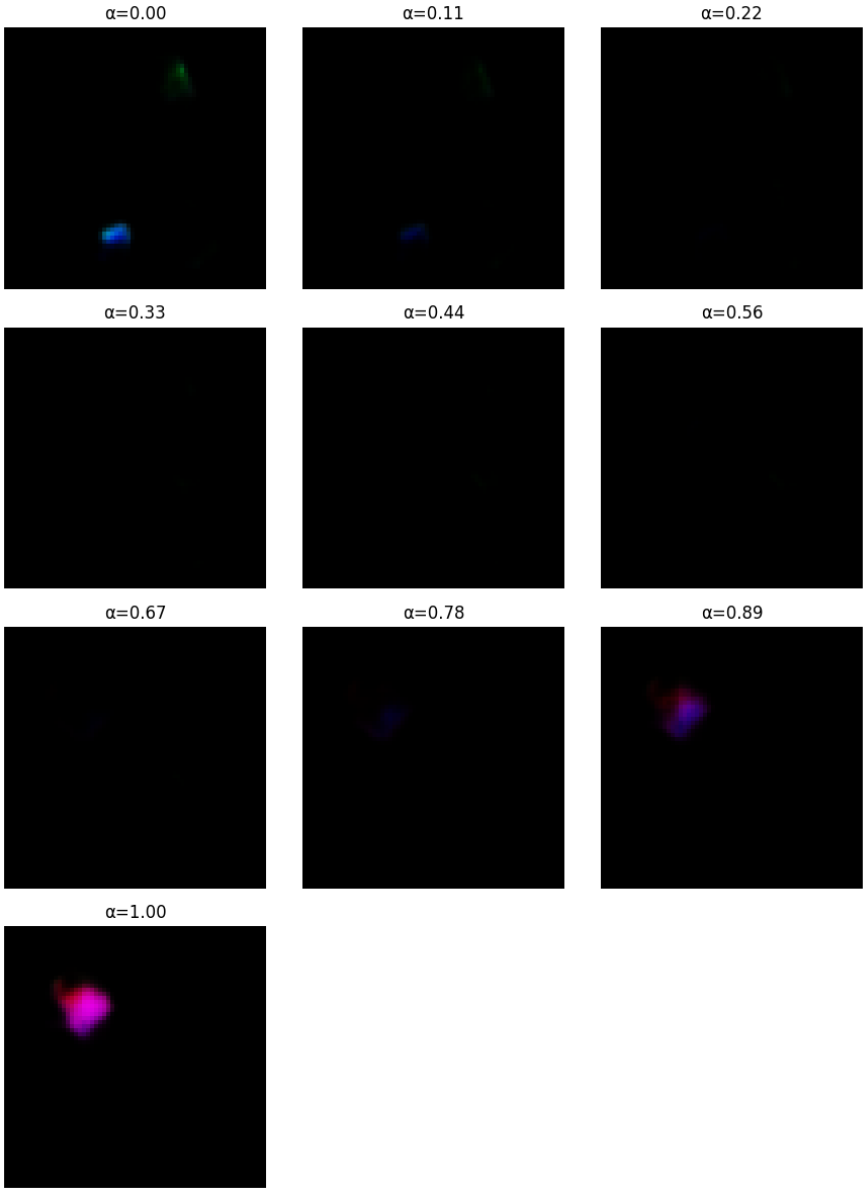
[3] C. P. Burgess et al., "Understanding Disentangling in β-VAE," *arXiv preprint arXiv:1804.03599*, 2018.

[4] A. Tacchetti et al., "Multi-Object Datasets," *https://github.com/addtt/multi-object-datasets*, 2019.

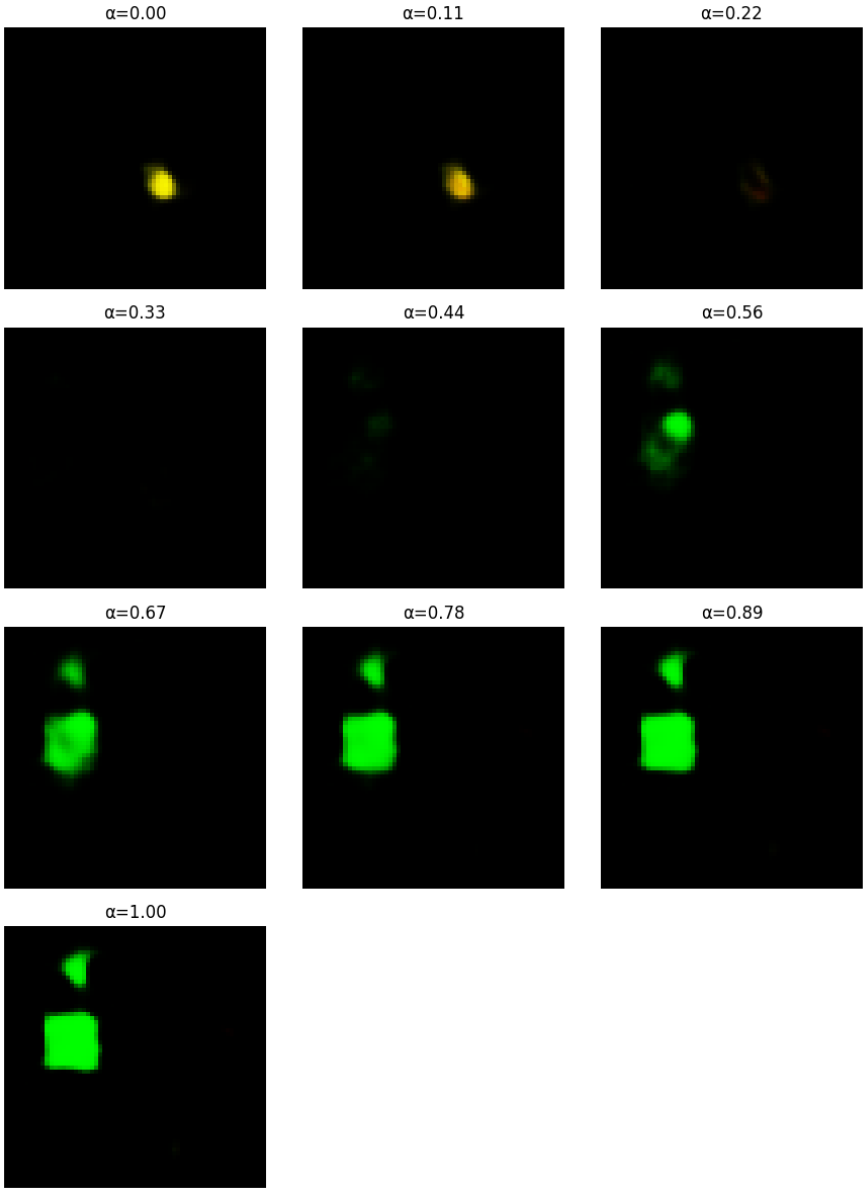
B=0.5, C=0.5~25

Test ELBO Loss: 35.8788

random



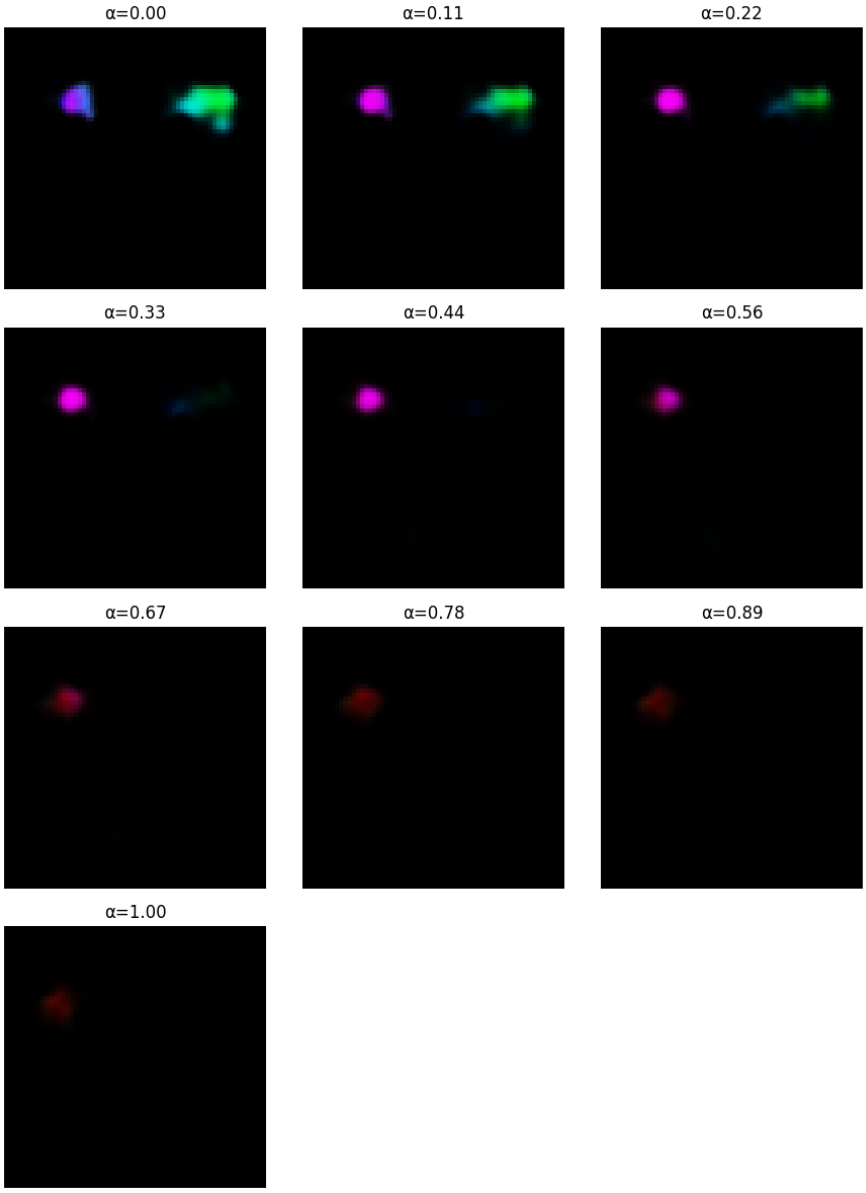
#Data based



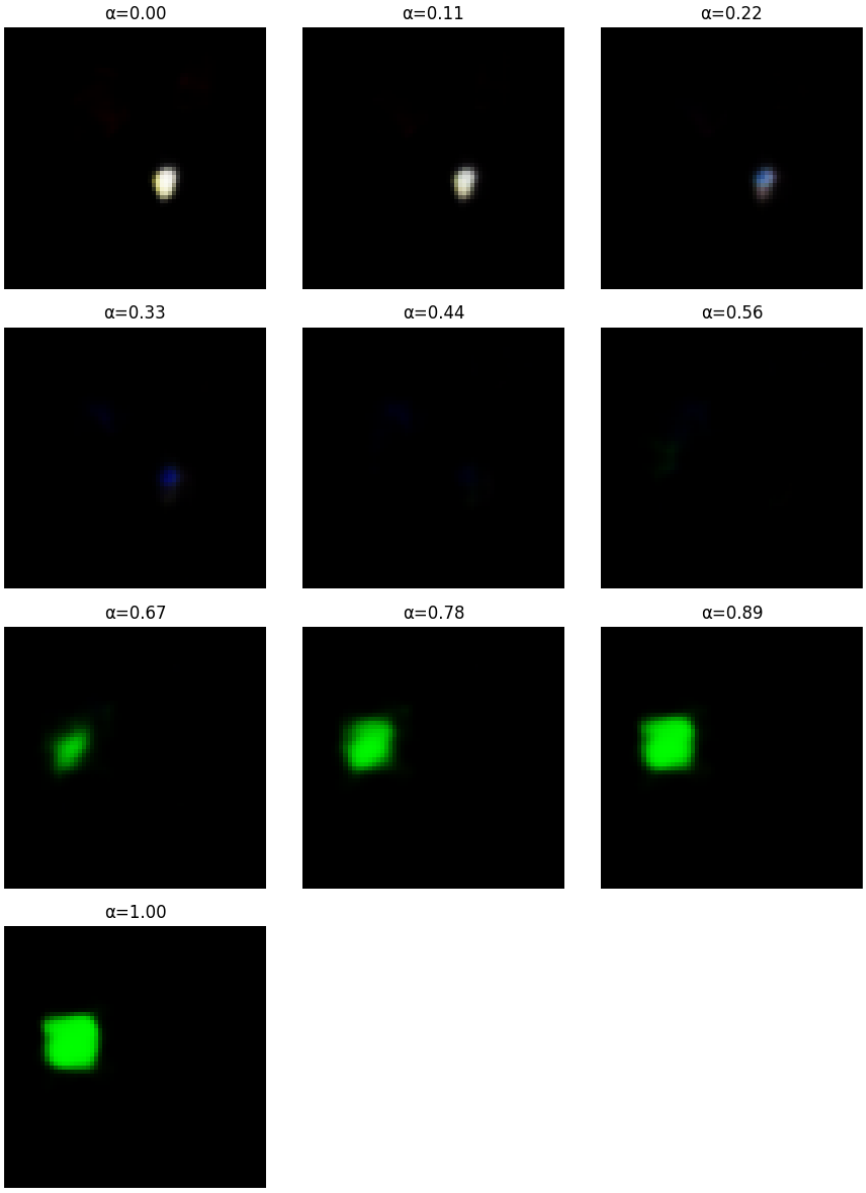
B=1, C=0.5~25

35.7178

random



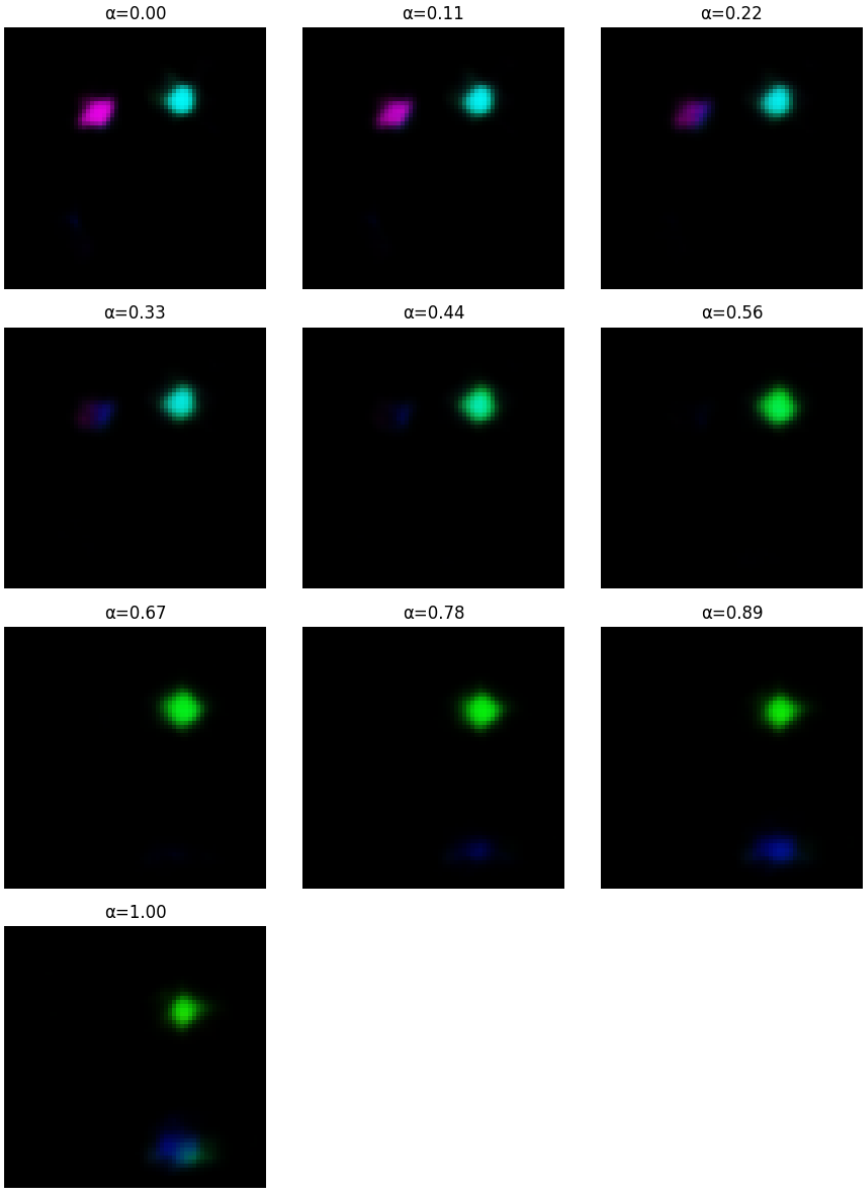
#data based



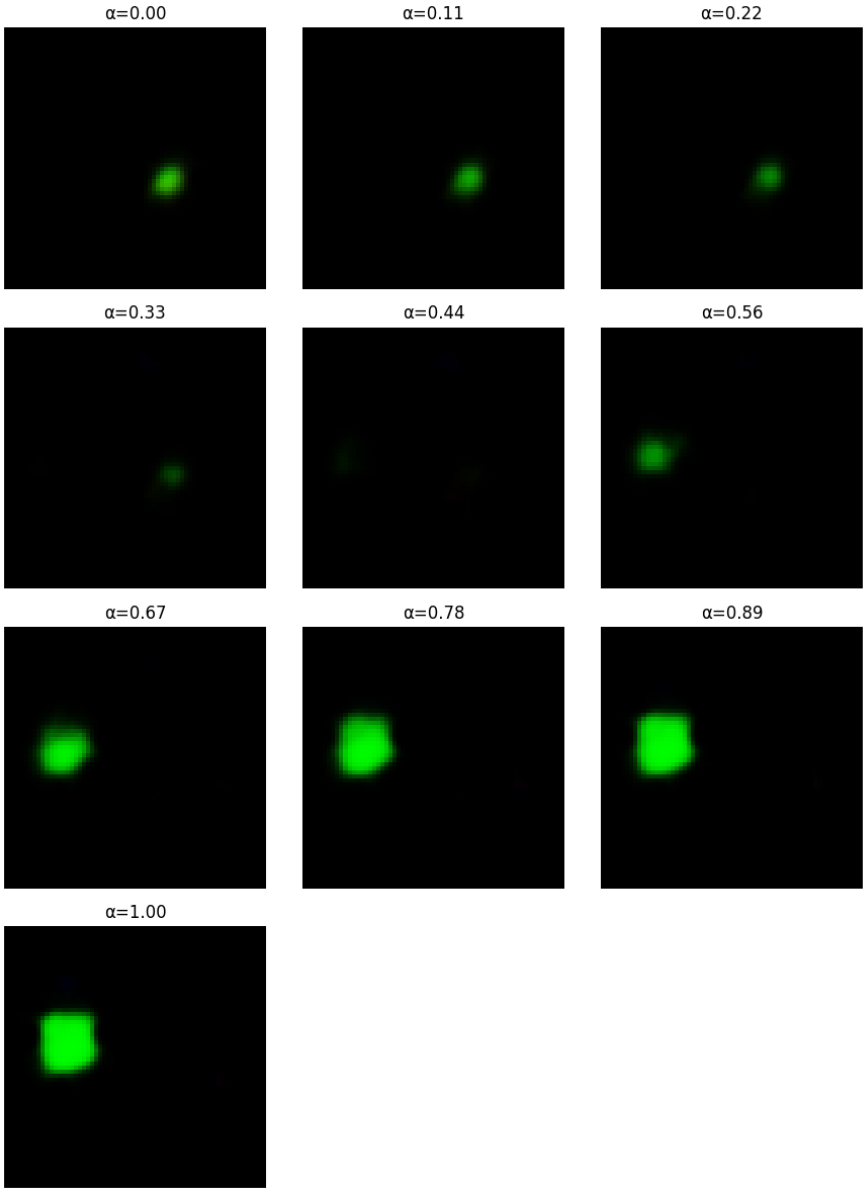
B=1.5, C=0.5~25

42.7203

random



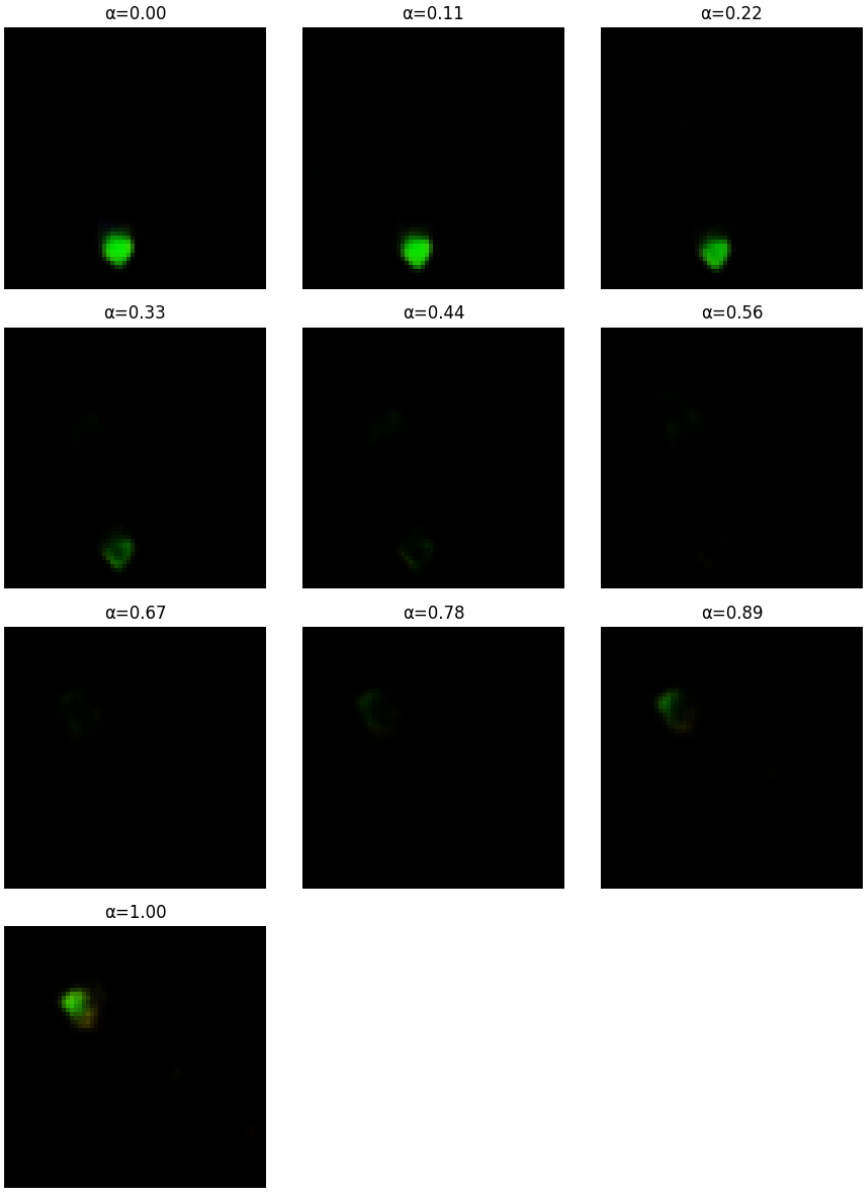
#data based



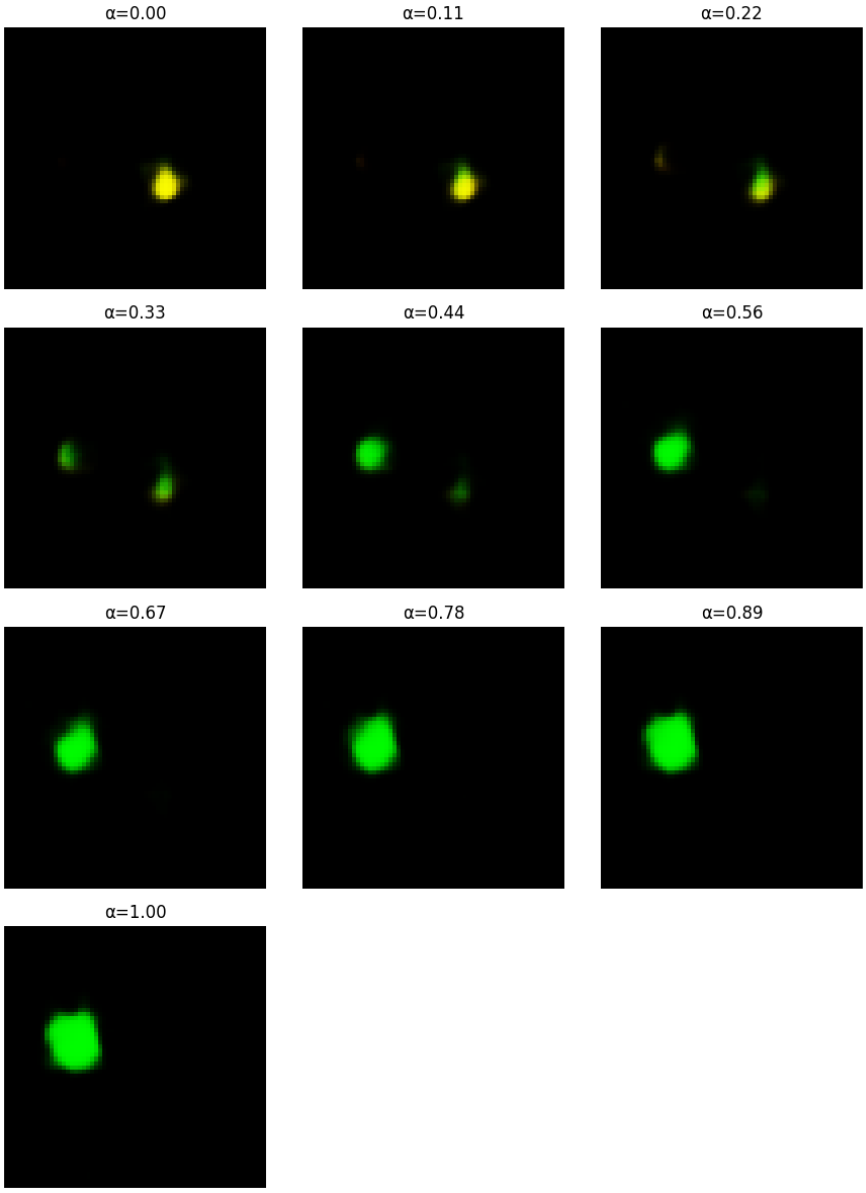
B=0.5, C=0

Test ELBO Loss: 34.1798

random



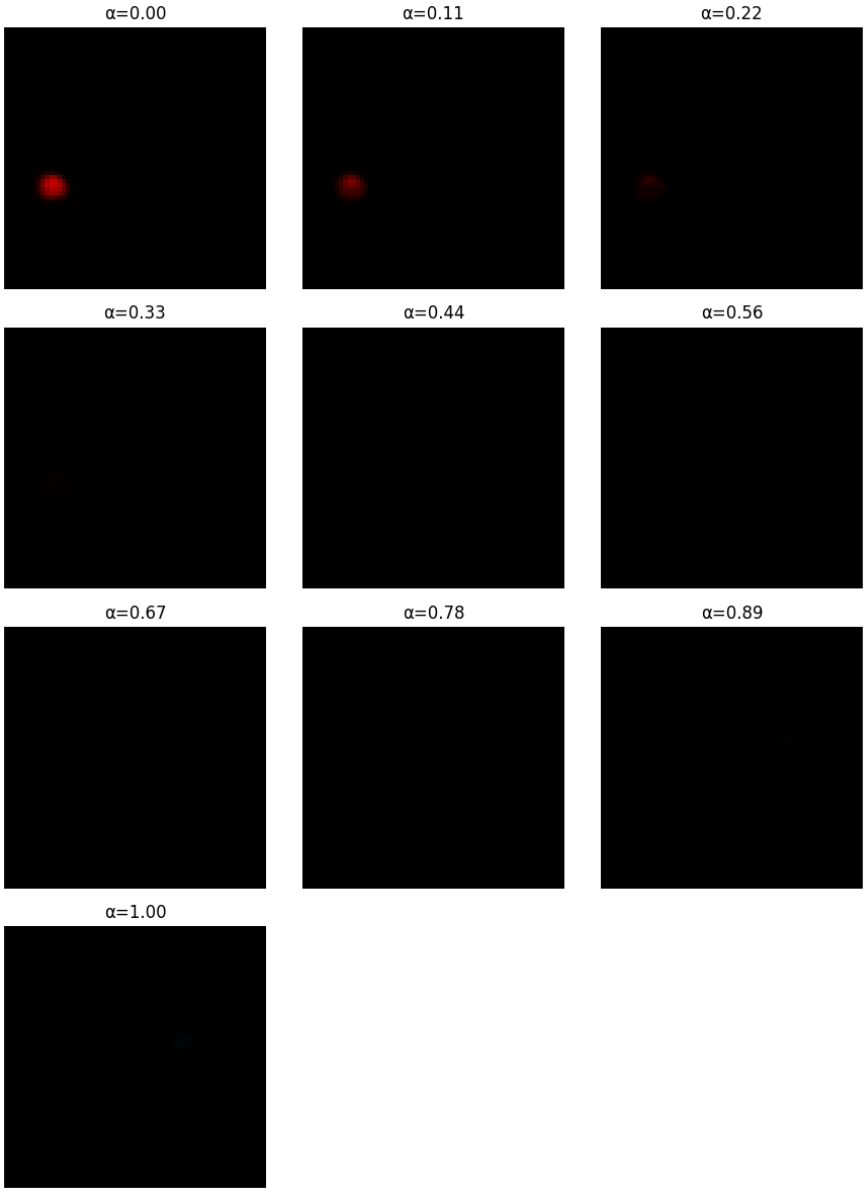
data based



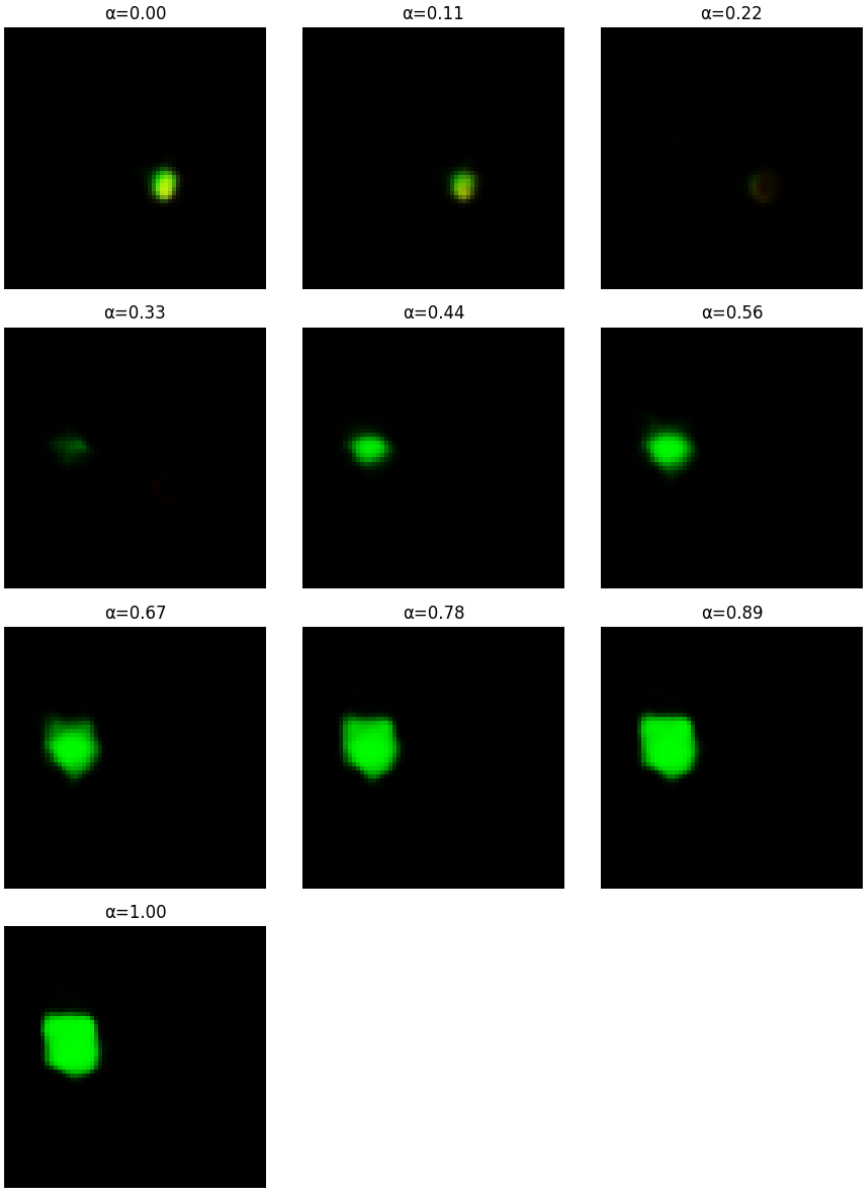
B=1 C=0

Test ELBO Loss: 35.0955

random



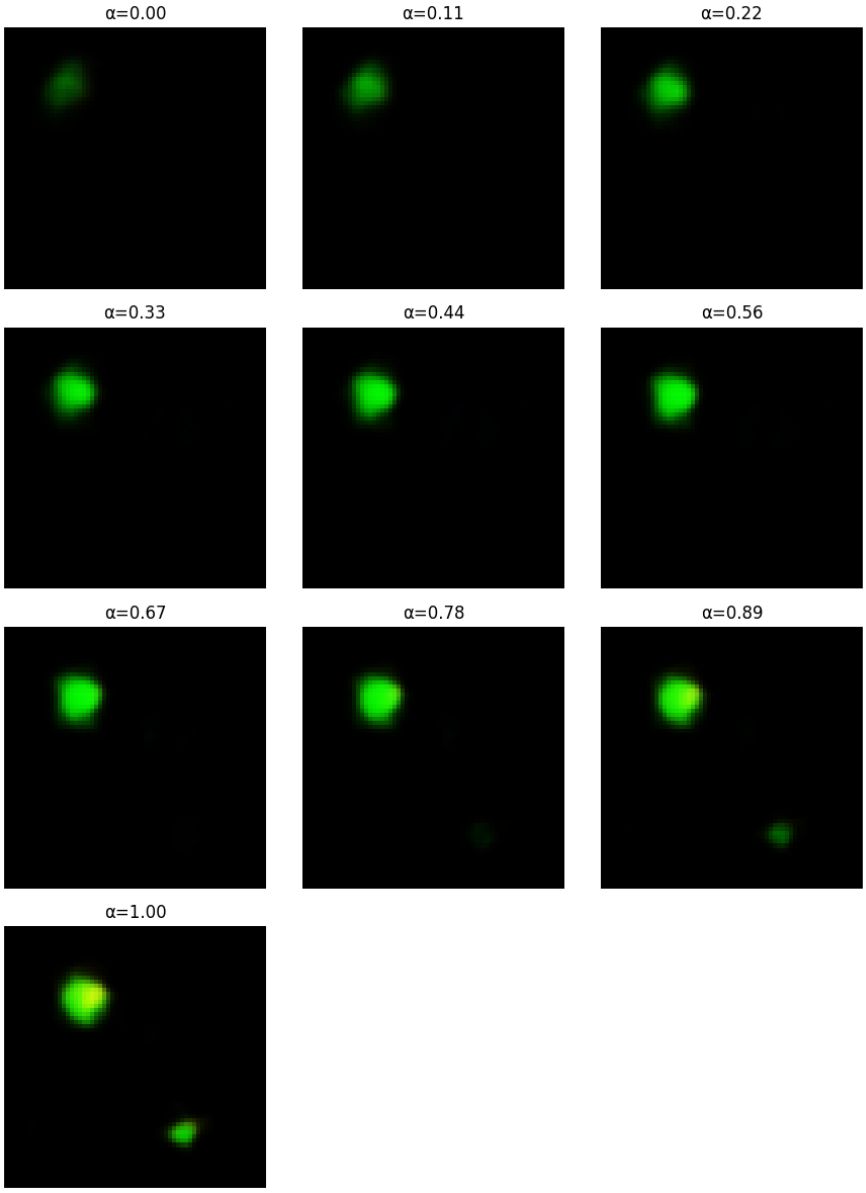
data based



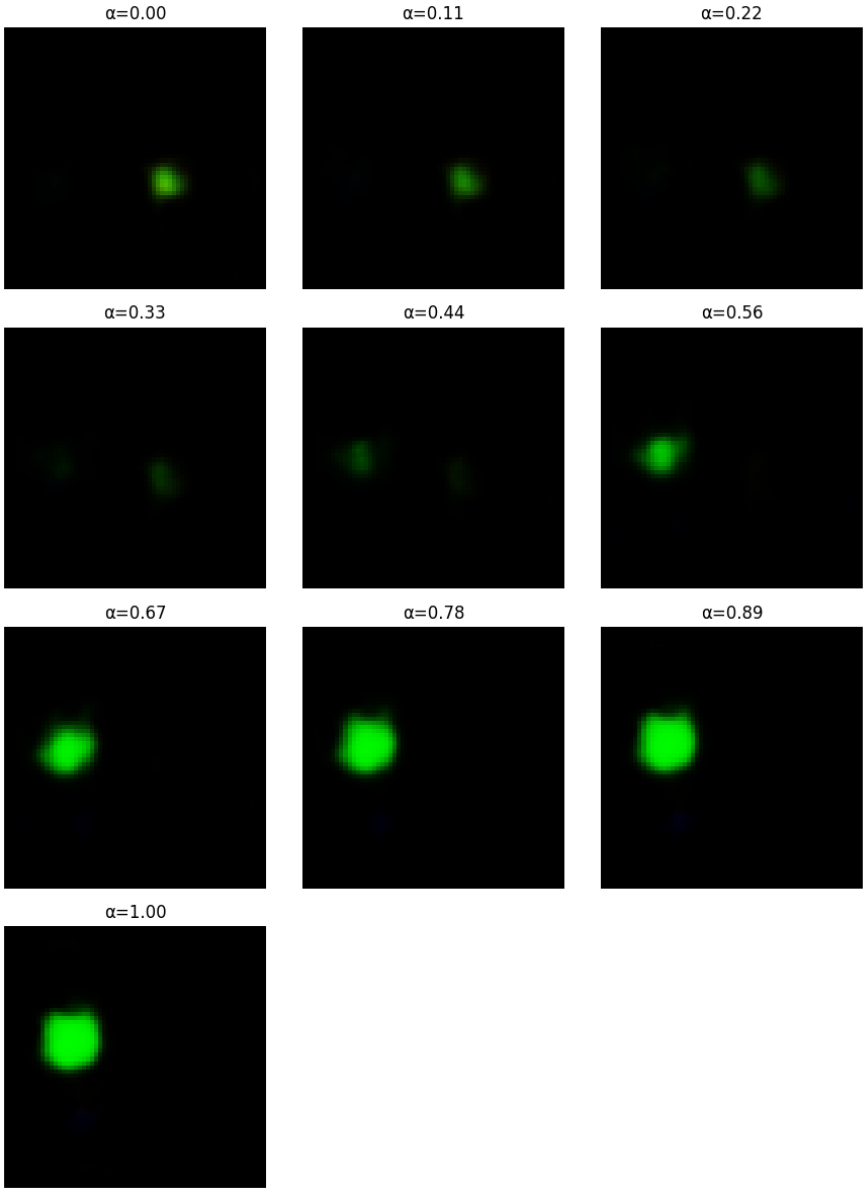
B=1.5 C=0

Test ELBO Loss: 38.5477

random



data based


α=0.00
α=0.11
α=0.22
α=0.33
α=0.44
α=0.56
α=0.67
α=0.78
α=0.89
α=1.00