Text-based Image Editing for Food Images with CLIP

Kohei Yamamoto Keiji Yanai
The University of Electro-Communications, Tokyo, Japan
{yamamoto-k,yanai}@mm.inf.uec.ac.jp

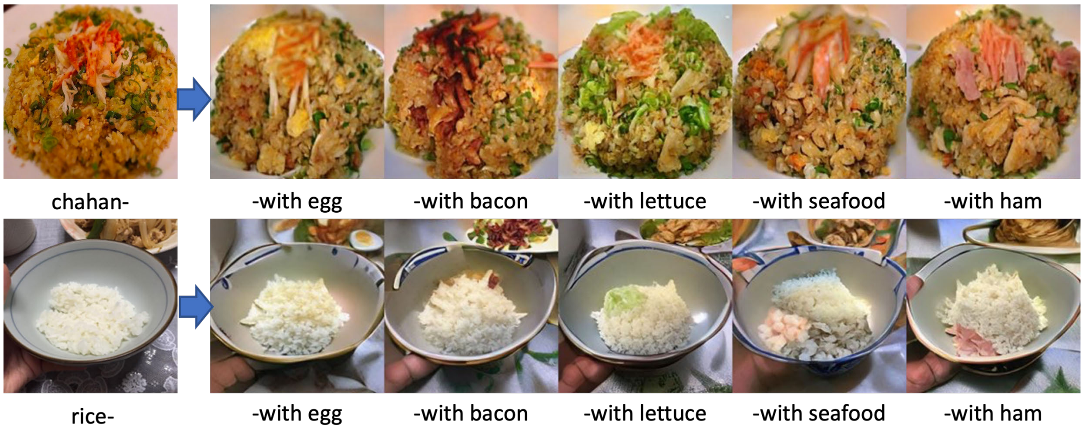

Figure 1: Example results of food image manipulation by texts. The leftmost column showed the original input images: chahan (fried rice in Japanese) and steamed rice. The second to sixth columns from the left showed the manipulated images by VQGAN-CLIP. The prompt used in each manipulation combined the food name and "with" a topping name. For example, the two images in the second column are generated with the prompts, "chahan with egg" and "rice with egg" respectively.

ABSTRACT

Recently, the large-scale language-image pre-trained model, such as CLIP, has drawn much attention due to its remarkable ability for various tasks, including classification and image synthesis. The combination of CLIP and GAN can be used for text-based image manipulation and text-based image synthesis.Several models of a combination of CLIP and GAN have been proposed so far. However, their effectiveness in the food image domain has not been examined comprehensively yet. In this paper, we reported the results of the experiments on text-based food image manipulation using VQGAN-CLIP and discussed the possibility of food image manipulation by texts.

CCS CONCEPTS

• **Computing methodologies** → **Image manipulation**; **Machine learning approaches**.

KEYWORDS

text-based image manipulation, food image manipulation, language-image pre-training model, CLIP

ACM Reference Format:
Kohei Yamamoto Keiji Yanai. 2022. Text-based Image Editing for Food Images with CLIP. In *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management (MADiMa '22), October 10, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3552484.3555751

1 INTRODUCTION

With the development of smartphones and social media, people have posted various photos on the Internet. Among them, one of the most frequently posted photos is meals. Beautiful, enormous, and eccentric meal photos become a topic easily on social media throughout the year. Some restaurants post many good-looking pictures to increase their sales by becoming a trend on social media. Taking a picture that looks delicious or novel requires many trial-and-error processes. After taking photos, they often edit them using advanced image editing software. Such software requires a high level of manipulating skill or knowledge.

The technology for editing images with deep neural networks has developed remarkably in computer vision, starting with GAN [1] in 2014. In this evolution, image editing using natural languages, such as ManiGAN [2] and StyleCLIP [3], has attracted attention as a new way to edit images because they do not require special skills or knowledge for editing. However, most of these models have not been applied to food images, mainly to human faces and animal images.

Moreover, the large-scale language-image pre-trained model, such as CLIP, has recently drawn much attention due to its remarkable zero-shot ability for various tasks, including classification and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MADiMa '22, October 10, 2022, Lisboa, Portugal
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9502-1/22/10…$15.00
https://doi.org/10.1145/3552484.3555751

image synthesis. The combination of CLIP and GAN can be used for text-based image manipulation and text-based image synthesis. Several models combining CLIP and GAN have been proposed, such as StyleCLIP and StyleGAN-NADA [4]. These methods can manipulate images with texts without training in a manipulation model. It is possible because CLIP was trained with four hundred million pairs of texts and images, and it has any knowledge of the relation between language and vision.

However, their effectiveness in the food image domain has not been examined comprehensively yet. This paper examines the possibility of text-based food image manipulation with many experiments. As an image manipulation method, we used VQGAN-CLIP [5]. As a result, we confirmed the effectiveness of text-based image manipulation using CLIP in the food domain.

2 RELATED WORK

There are two main types of natural language image editing models. One is a model which learns image-text pairs from scratch. The other is a model which uses pre-trained visual language models.

ManiGAN [2], which learns image-text pairs, contains a new text-image affine combination module and a detail correction module. This GAN generates converted images to specified colors or textures by text. A detail correction module could enhance the performance of maintaining irrelevant parts while editing details. TediGAN [6] is the model that used a pre-trained StyleGAN [7]. It has a similarity module, which learns the similarity between images and texts by mapping them to the same latent space. Using a StyleGAN trained on face images, TediGAN does not require GAN training time for the generator. However, this GAN restricts the face domain for the generated images. Models learning image-text pairs from scratch require a large amount of training time and images with text, limiting the edited type of images and manipulation.

The recent editing models often use the text encoder and image encoder of pre-trained visual-language models. In particular, CLIP [8] is the most common model as a pre-trained visual-language model. CLIP is trained on 400 million image-text pair data collected from the Internet. While models trained from scratch have limited training and narrow linguistic-visual features, CLIP has a sizeable amount of training and comprehensive linguistic-visual features. Therefore, it has been applied to various computer vision tasks, such as image classification, detection, segmentation, VQA, and image synthesis.

StyleCLIP [3] proposed three methods of editing the latent space for manipulating images by combining StyleGAN, a typical GAN in image generation, and CLIP. Several papers studied image editing by manipulating the latent space of StyleGAN. However, those were learned in semantic supervision or required human guidance. In this paper, CLIP automates such guidance. Paint by Word [9] is a part image editing model that combines StyleGAN and CLIP using masks. There are few studies to change a part of the image and keep the background. This model can edit the specific part in the mask by editing the latent code w of StyleGAN from the real images. However, this study uses StyleGAN and BigGAN [10], which are specialized GANs for bedrooms and birds or generic GANs, and not specialized for meals. In our study, we used $\hat{z}$ of trained VQGAN [11] in the meal domain.

The conclusive purpose of our study is to create an image editing model specialized for food by images and texts. Therefore, we used VQGAN-CLIP [5] and trained them on a set of meal images and texts. We also examined the mask function that manipulates only a part of the image for convenient editing as they thought.

3 METHODOLOGY

3.1 Image Manipulation Model

The model we used for food image editing is VQGAN-CLIP. By using VQGAN for the image generation part, it can control the image by a grid. Moreover, It will learn the vocabulary of image components using CNN [12] and their composition using Transformer [13]. Additionally, this model can generate high-quality images. CLIP could compute any linguistic-visual features and similarities between images and texts with high accuracy. In conventional image editing models, model architecture often fixed the text to the grammatical form for training. However, CLIP, trained data from the Internet, allows for various grammatical forms and can deal with ambiguous texts. This study used CLIP because natural text editing requires understanding this ambiguous text.

We examined VQGAN-CLIP, which trained VQGAN and CLIP on meal images instead of general datasets such as ImageNet, whether that is robust to meal features or not.

3.2 Architecture

The architecture of VQGAN-CLIP is shown in Figure 2.

First, the input image $x \in \mathbb{R}^{3\times H\times W}$ is resized to get a resized image $x_{Resize} \in \mathbb{R}^{3\times H'\times W'}$. Then, the resized image x_{Resize} is input to the encoder of VQGAN to generate the initial latent vector $\hat{z} \in \mathbb{R}^{n_z\times h\times w}$. Note that n_z is the number of dimensions of the VQGAN's codebook. Next, the latent vector $\hat{z}$ is input to the decoder of VQGAN, and the output image $\hat{x} \in \mathbb{R}^{3\times H'\times W'}$ and the input prompt t are input to CLIP encoders. These give the image token I and the text token T, and CLIP compute the loss. Then, the loss function updates the latent vector $\hat{z}$ by the gradient descent method. Finally, the latent vector $\hat{z}$ is clamped between the maximum and minimum values in the VQGAN's codebook, and the latent vector $\hat{z}$ is updated. Thus, the updated latent vector $\hat{z}$ is input to the decoder of VQGAN again, input to CLIP calculating the loss, and updated the latent vector $\hat{z}$ repeatedly.

Note that the latent vector $\hat{z}$ of VQGAN is a vector n_z assigned to each square grid $h \times w$. The model can restrict the editing range by using this. The formula1 shows the gradient calculation of the latent vector $\hat{z}$ by using $z_{mask} \in \mathbb{R}^{h\times w}$. One in the mask z_{mask} means a manipulatable grid, and zero means not a manipulatable image. In computing the gradient of the latent vector $\hat{z}_{grad} \in \mathbb{R}^{n_z\times h\times w}$, this study computes their element-wise product so that the gradient of the unchanged grid becomes zero. If we have a mask image that is the same size as the input size, the mask image is scaled down and transformed to fit the grid mask z_{mask}.

$$\hat{z}_{grad} \leftarrow \hat{z}_{grad} \odot z_{mask} \tag{1}$$

3.3 Loss Function

The manipulating model calculated the loss of equation 2. The whole loss is the sum of CLIP loss $\mathcal{L}_{CLIP}$ and image loss $\mathcal{L}_{img}$. This

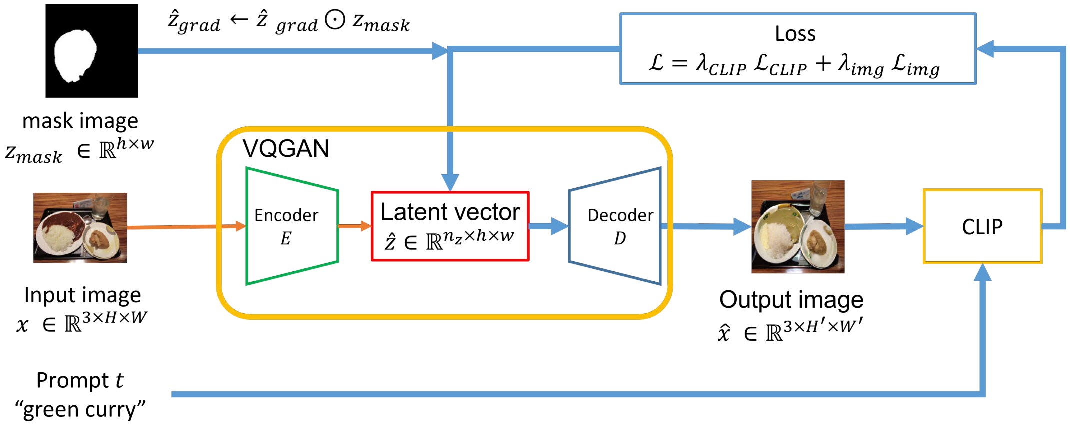
$\hat{z}_{grad} \leftarrow \hat{z}_{grad} \odot z_{mask}$
Loss
$\mathcal{L} = \lambda_{CLIP}\,\mathcal{L}_{CLIP} + \lambda_{img}\,\mathcal{L}_{img}$
mask image
$z_{mask} \in \mathbb{R}^{h \times w}$
VQGAN
Encoder
E
Latent vector
$\hat{z} \in \mathbb{R}^{n_z \times h \times w}$
Decoder
D
CLIP
Input image
$x \in \mathbb{R}^{3 \times H \times W}$
Output image
$\hat{x} \in \mathbb{R}^{3 \times H' \times W'}$
Prompt t
"green curry"

Figure 2: The VQGAN-CLIP architecture for food image manipulation. We trained VQGAN and CLIP with food datasets for specialization. VQGAN encoder makes a latent vector from an input image. From the latent vector, the VQGAN decoder makes an initial edited image. After that, CLIP calculates the losses along with the prompt. That loss optimizes the latent vector. This model repeated iterations to get the edited image. Optionally, we can use a mask when it optimizes.

study set the λ_{CLIP} and λ_{img} to one, respectively.

$$\mathcal{L} \quad = \quad \lambda_{CLIP}\mathcal{L}_{CLIP} + \lambda_{img}\mathcal{L}_{img} \tag{2}$$

Following equation 3 and equation 4 shows CLIP loss and image loss. CLIP loss and the image loss used the spherical distance loss from VQGAN-CLIP implementation. The spherical distance loss works as almost the cosine similarity between an image token and a text token. This loss can also calculate between an initial and generated image token. The image token I, I_{img} is the output of the CLIP image encoder for the generated image $\hat{x}$ and the resized image x_{Resize}. The text token T is the output of the CLIP text encoder for input prompt t.

$$\mathcal{L}_{CLIP} \quad = \quad 2\arcsin^2\left(\frac{I - T}{2}\right) \tag{3}$$

$$\mathcal{L}_{img} \quad = \quad 2\arcsin^2\left(\frac{I - I_{img}}{2}\right) \tag{4}$$

3.4 Food Image Datasets

Table 1 shows the statistic on the food image dataset we used in the experiments. To specialize in meals, VQGAN was fine-tuned to extract food features by three meal datasets with a different number of meal categories and images. On the other hand, CLIP was fine-tuned with Recipe1M [14], which includes pairwise text. This study used only the training and evaluation datasets of Recipe1M for training VQGAN and CLIP and the test dataset of Recipe1M for measuring metrics.

3.5 Prompts for Training CLIP Model

In order to determine whether to use the pre-trained CLIP model, we evaluated the no-pre-training model "title_NoPretrain" and the pre-trained model "title", which were trained with recipe title prompts. Then, we examined the learning prompts for transfer learning to use the pre-trained model. The Table 2 shows the list of learning prompts with the recipe title as "**some_title**" and the recipe material information as "**ingredients**".

Table 1: The list of food datasets. Regarding Recipe1M, we used both training and validation sets for training VQGAN.

datasets name	number of categories	number of images
Magical Rice Bowl [15]	10	80,408
Foodx251 [16]	251	158,846
Food500 [17]	500	399,726
Recipe1M [14](Train,Valid)	-	753,251

Table 2: The list of prompts for training CLIP.

abbreviation	prompts for training	pre-train
title_NoPretrain	**some_title**	×
title	**some_title**	○
ingredients	**ingredients**	○
ingredients_title	**ingredients** + ' are ingredients in ' + **some_title** + ' .'	○
APhotoOfA	'A photo of a ' + **some_title** + ' .'	○
APhotoOfA_ATOF	'A photo of a ' + **some_title** + ' , a type of food .'	○

These prompts are based on CoOp [18]; adding "a" before the class token improves the classification accuracy by over 5%. Most times, adding "a photo of a" before the text improves the classification accuracy. Furthermore, adding context related to the task make a significant improvement. For example, adding "a type of flower" increased the classification accuracy of the flower image dataset. For the training prompts of CLIP, we also added these prefixes and suffixes to examine the difference in performance.

Recipe1M contains recipe information such as titles and ingredients. This study used the titles as the main prompt and the material information as the additional prompt. The training prompt "ingredients" used only the ingredients information. The "ingredients_title" which used a combination of the dish name and the ingredients name used title as "**ingredients**" and ingredients as "**some_title**".

4 EXPERIMENTS

4.1 Overview

First, we compared the prompts for image editing. The compared items by prompt are these.

(1) Differences by the calling way
(2) Differences between inside and outside of the learning domains
(3) Differences by taste adjectives
(4) Differences by toppings

Then, we showed the differences when VQGAN trained on different datasets, followed by the differences because of the prompts during CLIP training. Finally, we present a quantitative evaluation of VQGAN using metrics.

4.2 Evaluation Metrics

Quantitative evaluation of GAN in this study used Inception score (IS) [19], Freshet initiation distance (FID) [20], and Kernel-Inception distance (KID) [21].

IS is the Kullback-Leibler divergence (KL-divergence) between the conditional label distribution $p(y|x_i)$ and the surrounding label distribution $p(y)$ for the generated data. This metric marks higher the more significant the diversity of images and the more manageable the images are to identify. In general, higher IS is suitable for image editing. The IS is derived by averaging the KL-divergence used number of images$i$. It describes it as the equation 5.

$$\mathrm{IS} = \exp(\frac{1}{N}\sum_{i}\mathrm{KL}(p(y|x_i)||p(y))) \tag{5}$$

FID is one of the famous metrics to measure the feature distance between a real image and a generated image. This metric is also used to evaluate the quality of GAN. Let $\boldsymbol{m_w}$ and $\boldsymbol{C_w}$ denote the mean and covariance matrix of feature vectors of the real image, and $\boldsymbol{m}$ and $\boldsymbol{C}$ denote the mean and covariance matrix of feature vectors of the generated image, respectively, FID is defined by the equation 6. Let Tr be the trace of the matrix. The lower the FID value is, the higher the image quality is.

$$\mathrm{FID} = ||\boldsymbol{m} - \boldsymbol{m_w}||^2 + \mathrm{Tr}\big(\boldsymbol{C} + \boldsymbol{C_w} - 2\sqrt{\boldsymbol{C}\boldsymbol{C_w}}\big) \tag{6}$$

KID is the dissimilarity calculated using Maximum Mean Discrepancy (MMD) with (f_{real}, f_{fake}) samples drawn independently from different distributions. KID is defined as in equation 7.

$$\mathrm{KID} = \mathrm{MMD}(f_{real}, f_{fake})^2 \tag{7}$$

We calculated these metrics between 50,000 real images on the test dataset from Recipe1M and 50,000 reconstructed images from trained VQGAN. Those images were resized 256×256. We calculated these metrics between 50,000 real images on the test dataset from Recipe1M and 50,000 reconstructed images from trained VQGAN. Those images were resized 256 × 256. We measured IS by using torch-fidelity [1]. We used the clean-fid [2] to measure the FID. KID is also measured by using torch-fidelity.

In addition, evaluation indiced for CLIP used median rank(medR), Recall(R@1, R@5, R@10), and CLIPScore [22]. medR is the median search rank. Recall is the percentage of the search results within the first, fifth, and tenth ranks. CLIPScore evaluates the quality of the generated caption candidates. The loss may be high using CLIP as a loss function even though the caption is a match. Similarly, the loss may be low even if the prompts are inappropriate. CLIPScore is used to evaluate how much the text match. With $w = 2.5$, c as the caption token, and v as the image token, it is calculated as follows.

$$\mathrm{CLIPScore} = w * max(cos(c, v), 0) \tag{8}$$

We used random 10k Recipe1M test data for evaluating CLIP. OpenCLIP [3] was used for CLIP training and calculating medR and Recall. The authors' implementation [4] used for measuring CLIPScore.

4.3 Implementation Details

We fine-tuned the following five VQGAN in this study.

(1) trained ImageNet-1024 model
(2) trained ImageNet-16384 model
(3) trained Magical Rice Bowl model for 59 epochs
(4) trained Foodx251 model for 62epochs
(5) trained Food500 model for 12 epochs

Each image was resized to a square of 256 × 256, and the output resolution was set to the same size. The size of the codebook was 256, divided into a grid of 16 × 16, and the latent vector was $\hat{z} \in \mathbb{R}^{256\times16\times16}$. CLIP was trained using openCLIP [3], an open source version of CLIP. OpenCLIP was trained on the training and evaluation datasets of Recipe1M with training prompts of Table 2. CLIP trained in 32 epochs with ResNet50 [23] as the backbone. The optimization function in the training part was AdamW [24], which set the batch size to 64, learning rate to 0.001, and weight decay factor to 0.1. The optimization function in the image generation part used Adam, which set the step size to 0.05. We iterated image optimization 1000 times for image editing of one image.

5 EXPERIMENTAL RESULTS

We show the output result from Figure 3 to Figure 12. The required time for image editing was about 4 to 6 minutes per image.

5.1 Prompt Differences

We compared the prompts for editing images using VQGAN-CLIP. In this5.1 section, we used VQGAN trained on the Magical Rice Bowl dataset and ViT-B/32's pre-trained CLIP. We used VQGAN trained on the Magical Rice Bowl dataset to clarify whether there is a difference between the trained and untrained images in Figure 4.

Figure 3 compared how meals are called. Magical Rice Bowl dataset has only ten categories of Japanese dishes. We compared "adjective + English meal name", "adjective + Japanese meal name", and "adjectives only". In all the output images in Figure 3, the color of each prompt is visible in various places. With adjectives only, the color changed in a large range, and its change was also flat like painted, instead of the color change by the ingredients in adjective only. Comparing the "adjective + English name" and the "adjective + Japanese meal name", the "adjective + Japanese name" looked more natural. The English meal name might be perceived as a cooking

[1] https://github.com/toshas/torch-fidelity
[2] https://github.com/GaParmar/clean-fid
[3] https://github.com/mlfoundations/open_clip
[4] https://github.com/jmhessel/clipscore

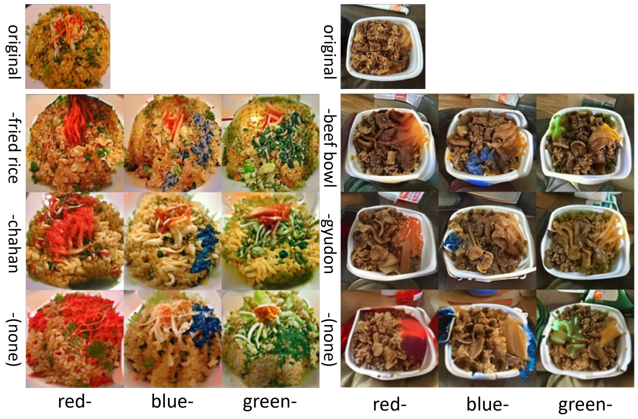

Figure 3: Differences in a calling way. We compared “adjective + English meal name”, “adjective + Japanese meal name”, and “adjectives only”. Adjectives (colors) used as prompts are listed at the bottom and the meal name on the left. For example, the leftmost column has the prompts “red fried rice”, “red chahan”, and “red” from the top.

process rather than a food name if it is separated into two words, such as “fried rice”.

Next, we showed the difference between included and excluded meals in the Magical Rice Bowl dataset. Here, we chose gyudon (beef bowl), kaisendon (seafood bowl), and yakisoba(stir-fried noodles) as images included in the Magical Rice Bowl dataset, and steak, pizza, and pasta as images not included. Figure 4 showed its results. When we input the prompt gyudon, objects in gyudon ingredients appeared. When the prompt was yakisoba, noodle-like objects were seen in both input images. In addition, we saw the color and shape of steak and the color of pizza, which did not include the ten meals in the Magical Rice Bowl dataset. Thus, there are some changes in all the output images. There is no significant difference between the meals included in the Magical Rice Bowl dataset and those not. The inclusion or exclusion in the training domain of VQGAN did not significantly affect the image edits. Therefore, these image changes were not caused by the training of VQGAN but depended on CLIP.

In addition, to observe the difference in the taste adjectives, we added typical taste adjectives to the meal names. The selected tastes are hot, sweet, salty, sour, bitter, and oily. Figure 5 showed the outputs when input prompts of taste adjectives. No significant change in appearance is observed in all images except for oily. From these outputs, we inferred that the input related to taste might not change much because of the weak visual meaning. To make these edits, we should add words such as spicy ingredients with another visual change.

Finally, we showed the output results using the prompts for adding toppings in Figure 1. Here, we considered the model to change the images by adding words of five ingredients: egg, bacon, lettuce, seafood, and ham, after adding the word “with”. As shown in Figure 1, outputs are visible five ingredients that do not exist in the categories of the Magical Rice Bowl dataset. It happened by linking with the large-scale pre-trained visual language model CLIP. We also found that there are cases editing different parts, such as the dish of the upper right part in “rice with egg”. Therefore, we examined a function to input masks.

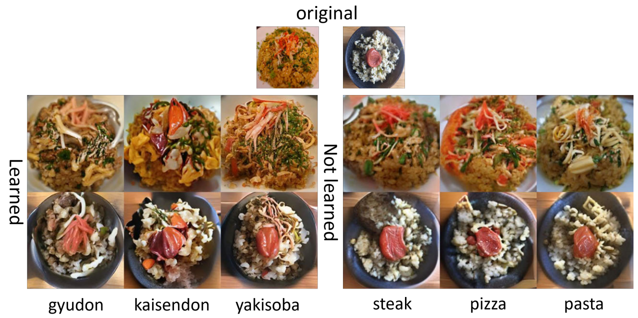

Figure 4: Differences between inside and outside of the learning domains on VQGAN. The left side is the domain within the learning, and the right side is the domain outside the learning. The input prompts are listed below. VQGAN’s learning domain or not does not make a significant difference in image editing.

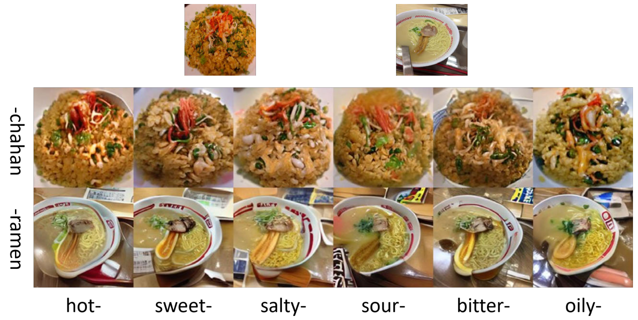

Figure 5: Differences by taste adjectives. The prompts used are listed below. Except for “oily”, the appearance did not change significantly.

The function of adding toppings is a feature of this image editing model. Hence, the following comparison by the VQGAN and CLIP models used the prompts about toppings.

5.2 Differences of Trained VQGAN

Figure 6 showed the differences among the models of trained VQGAN with food datasets. We used the shown VQGAN in the figure and the pre-trained ViT-B/32 CLIP.

There are differences in image editing in all outputs, but they are edited as the prompts. There are no significant differences in quality among all training models, only minor differences.

5.3 Differences of Trained CLIP

Figures 7 and 8 compared the training prompts when training CLIP with Recipe1M. We used the CLIP learned from the prompts shown in Table 2, except for ViT-B/32 pre-trained CLIP for the original CLIP. This section outputs used the ImageNet1024 pre-trained VQGAN.

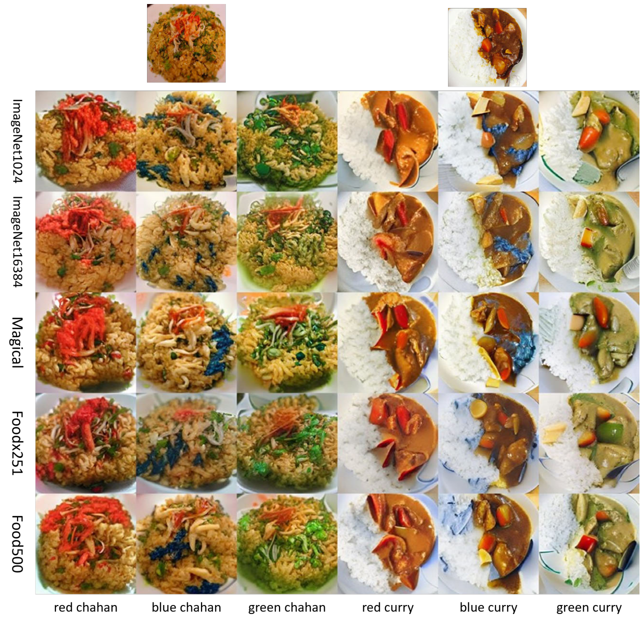

Figure 6: Differences in training datasets of VQGAN. The prompts are listed below, and the dataset used for training VQGAN is on the left. There are no significant differences in quality among all training models.

Comparing "title_NoPretrain", which was learned with the title from scratch CLIP, and "title", which was learned with title from the pre-trained CLIP, there was no apparent difference. Therefore, we used the pre-trained model for the other learning prompts. The outputs of "ingredients" and "ingredients_title" prompts that include ingredients have similar artifacts. In addition, those CLIP models could not manipulate images according to the prompts, and lower quality of image editing was observed. We can assume that learning CLIP with food ingredients does not improve the image quality. This conjecture implies that the material does not directly play a significant role in the image's appearance. We also found that training CLIP with meal images does not suppress GAN-specific artifacts in whole result images. As for "APhotoOfA" and "APhotoOfA_ATOF", both models were less disturbed, but "APhotoOfA_ATOF" was less corrupted overall outputs. "APhotoOfA_ATOF" was also more stable than "APhotoOfA". In Figure 7, we can see bacon, lettuce, and seafood on a different plate from the rice in the upper right of the image in "APhotoOfA_ATOF". Also, in Figure 8, we can see an egg, bacon, and ham in "APhotoOfA_ATOF". Finally, comparing the original CLIP and the whole trained CLIP, the original CLIP showed some changes along with the prompts, but the image quality was rough.

5.4 Difference with and without Mask

In this section, we showed the results using the mask. Figure 9 to 12 showed the results. These images used VQGAN trained on the Magical Rice Bowl dataset and pre-trained CLIP on ViT-B/32. In addition, in Figure 9, we manually created the square mask to cover the rice, while in other Figures 10 to 12, we used the mask from UECFoodPixComplete [25].

In Figure 9, the toppings were placed on top of the rice, and the surrounding background was unchanged with the mask. In

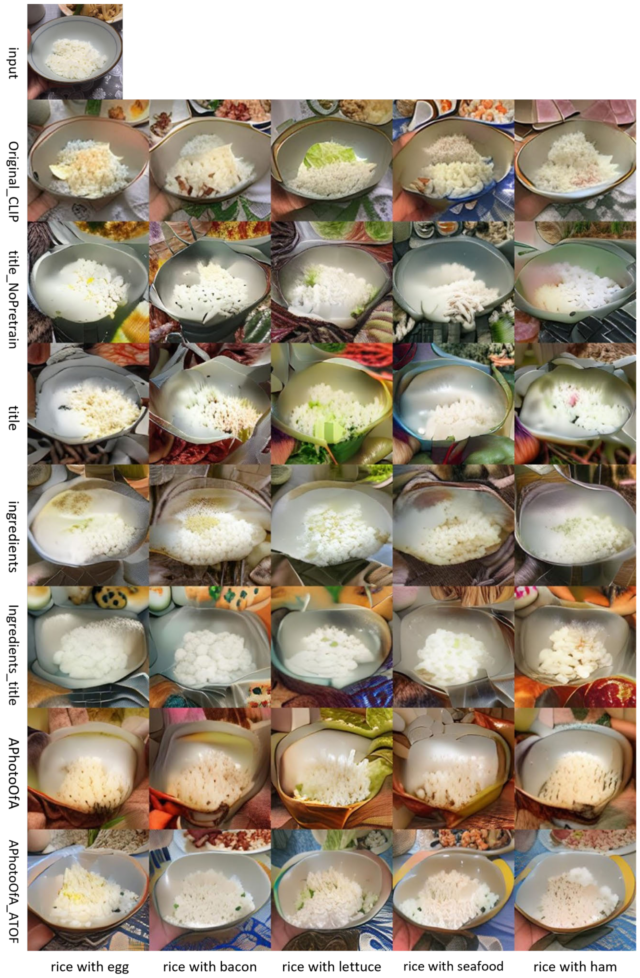

Figure 7: Difference by CLIP learning prompt on rice. The prompts are listed below, and the CLIP-learned prompts (see Table 2) are on the left. There was no obvious difference between "title_NoPretrain" and "title". The quality of image editing was lower for "ingredients" and "ingredients_title", higher for "APhotoOfA_ATOF" and "PhotoOfA".

Table 3: Reconstructed metrics of 50,000 meal images in VQ-GAN

50k	IS↑	FID↓	KID↓(×10^{-3})
ImageNet1024	**7.09 ± 0.10**	6.73	3.92 ± 0.46
ImageNet16384	7.05 ± 0.06	4.53	2.39 ± 0.36
Magical Rice Bowl	5.97 ± 0.06	7.15	3.51 ± 0.48
foodx251	6.15 ± 0.06	4.59	1.85 ± 0.31
food500	6.62 ± 0.05	**4.07**	**1.66 ± 0.29**

Table 4: Quantitative evaluation of CLIP

	image to text				text to image				CLIPscore↑
10k	MedR↓	R@1↑	R@5↑	R@10↑	MedR↓	R@1↑	R@5↑	R@10↑	
title_NoPretrain	21	0.105	0.278	0.381	22	0.109	0.278	0.384	**1.2866**
title	17	0.113	0.307	0.419	**17**	0.120	0.306	0.416	1.2270
ingredients	1022	0.008	0.030	0.050	827	0.006	0.024	0.041	0.4418
ingredients_title	1066	0.016	0.058	0.095	808	0.011	0.040	0.066	0.4176
APhotoOf	**16**	**0.117**	**0.310**	**0.424**	**17**	**0.121**	**0.308**	**0.422**	1.1517
APhotoOf_typeOfFood	17	0.107	0.299	0.415	**17**	0.108	0.302	0.412	1.2537

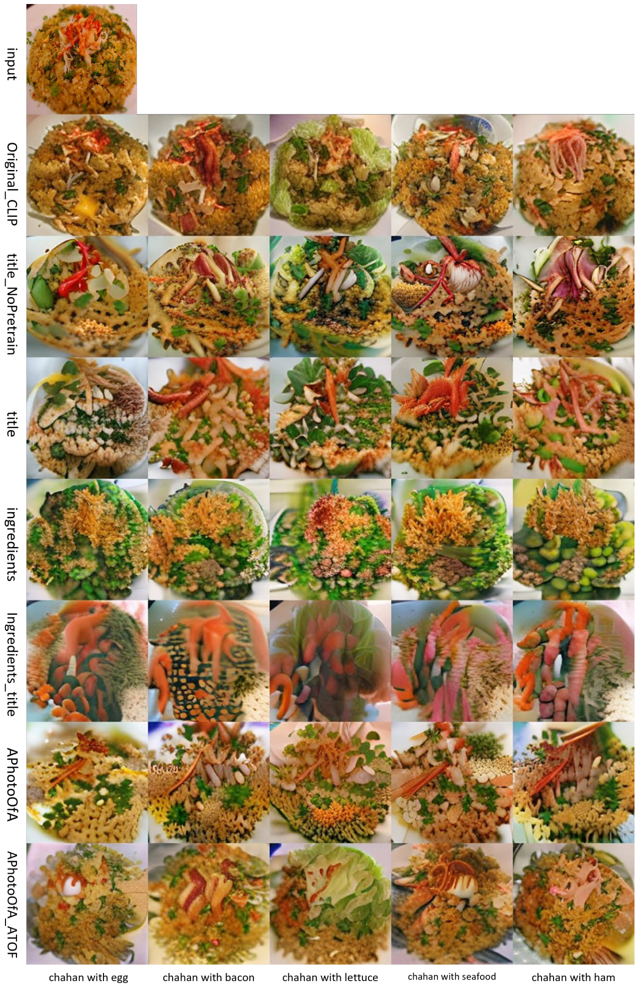

Figure 8: Difference by CLIP learning prompt on chahan. The prompts are listed below, and the CLIP-learned prompts are on the left. The result is almost the same as Figure7.

Figure 10, we used a circular mask got from the mask image from UECFoodPixComplete. We can confirm that model added the toppings to the curry. In Figure 11, we used a large rectangular mask got from UECFoodPixComplete. Toppings are even sometimes visible without the mask, but all the toppings are visible with the mask.

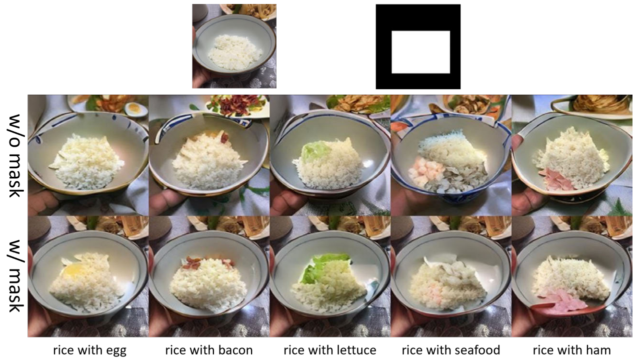

Figure 9: Differences of output by using the manually created grid mask on rice images. The small images at the top are the input image and the input mask. The prompts are shown at the bottom. The upper images are the output result without the mask, and the lower image is the output result with the mask. The results show that the background is preserved.

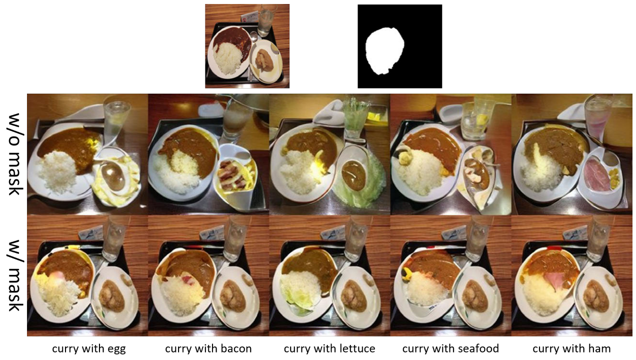

Figure 10: Differences of output by using the mask on curry images. The format is the same as in Figure9. Those results used a circular mask got from the UECFoodPixComplete mask image. The model was edited to add toppings to the mask.

In Figure 12, we used a large uneven mask like a circle from UECFoodPixComplete. The toppings are present even without masks, but with masks, toppings are generated obviously. Moreover, the model uses a mask not to change the spoon's shape.

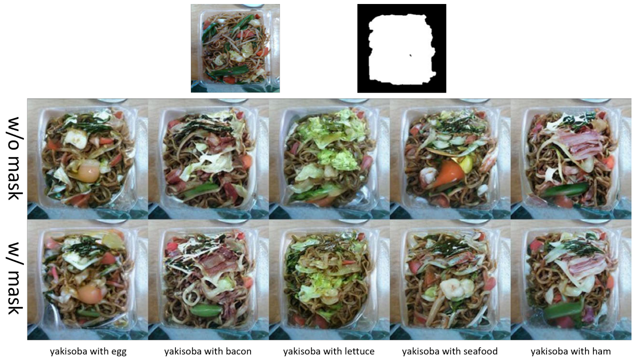

Figure 11: Differences of output by using the mask on yakisoba images. The format is the same as in Figure9. Toppings are even sometimes visible without the mask, but all the toppings are visible with the mask.

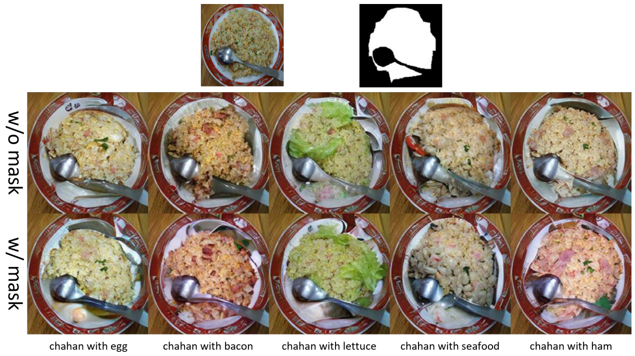

Figure 12: Differences of output by using the mask on chahan images. The format is the same as in Figure9. Those results used a large uneven mask that got from UECFoodPixComplete. The spoon shape is maintained by using a mask.

Using the mask, the model did not edit the image at the specified location, and the manipulated images were not corrupted. In addition, this model edited the non-masked area appropriately, and there was less image corruption near the mask's boundaries. When an edited range of the image is small, editing without a mask often results in not being edited well because manipulation attention is drawn to various locations. On the contrary, when the edited range of the image is extensive, editing is often performed even without a mask. However, as shown in Figure 12, a thing's shape, like the spoon, is maintained by a mask. Therefore, we can say that editing quality is better with a mask in all cases. Although there is a need to input a mask, the input of a mask image is a very effective method when we want to specify the editing points or when we want to make drastic editing.

5.5 Quantitative Evaluation of the Model

Table 3 showed the evaluation of VQGAN. VQGAN pre-trained on food500 had the lowest FID and KID, while VQGAN trained on the Magical Rice Bowl dataset had the highest FID. The FIDs of ImageNet16384 and foodx251 were comparable to those pre-trained on food500, but ImageNet16384 KID was a little far from food500. The IS comparison showed that ImageNet1024 was the best, which indicated the most recognizable and diverse, but food500 is not far behind, and overall, food500 is the highest quality model.

The differences among the training datasets of VQGAN were evaluated quantitatively and qualitatively in Table 3 and Figure 6. The differences among the training datasets in Figure 6 showed that the output images are not significantly different. Then, when we look at Table 3, we find that the food500 model has the highest accuracy. The quantitative evaluation showed a difference in the numerical values, but there is no significant difference in quality in image editing. It showed that qualitative evaluation by quantitative evaluation is complex. In image editing, such as this study, no standard evaluation index and benchmark has been formed, and quantitative evaluation does not directly evaluate the superiority of image editing. The establishment of a new quantitative evaluation index is required in image editing.

Table 4 showed the quantitative evaluation of CLIP. "title", "APhotoOF", and "APhotoOf_ATOF" are relatively dominant. "title_NoPretrain" had the highest CLIPScore, but may not be mature about the similarity between text and images. Figures 7 and 8 also showed the differences among the training prompts for CLIP. When models get a good quantitative evaluation, the output images also tend to be good. Therefore, unlike GAN quantitative evaluation, CLIP evaluation has the potential to provide qualitative evaluation through quantitative evaluation.

6 CONCLUSIONS

In this study, we examined the effectiveness of VQGAN-CLIP in editing images related to meals. In order to further specialize the model to meal images, we trained VQGAN on a meal image dataset and CLIP on a recipe dataset with various training prompts.

We compared the differences by editing prompt, training dataset for VQGAN, and training prompts of CLIP, using a mask or not. We found that the training dataset for VQGAN showed no significant differences in the output images. However, the evaluation index is better with the meal image dataset. "APhotoOfA_ATOF" has relatively fewer image corruptions and better scores in the quantitative evaluation among the CLIP training prompts. The function of mask images is a very effective tool for image editing.

We will consider adding a model that automatically inputs masks. We will also consider image generation models other than the VQGAN model, such as PPDM[26].

Acknowledgment This work was supported by JSPS KAKENHI Grant Numbers, 21H05812, 22H00540, 22H00548, and 22K19808.

REFERENCES

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc.of Advances in Neural Information Processing Systems*, 2014.

[2] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proc.of IEEE Computer Vision and Pattern Recognition*, pages 7880–7889, 2020.

[3] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *arXiv preprint arXiv:2103.17249*, 2021.

[4] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. In *arXiv*

preprint arXiv:2108.00946, 2021.

[5] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. *arXiv preprint arXiv:2204.08583*, 2022.

[6] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *Proc.of IEEE Computer Vision and Pattern Recognition*, pages 2256–2265, 2021.

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc.of IEEE Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *arXiv preprint arXiv:2103.00020*, 2021.

[9] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. In *arXiv preprint arXiv:2103.10951*, 2021.

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *arXiv preprint arXiv:1809.11096*, 2018.

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proc.of IEEE Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

[12] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Proc.of Shape, Contour and Grouping in Computer Vision*, pages 319–345. Springer, 1999.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc.of Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[14] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning cross-modal embeddings for cooking recipes and food images. In *Proc.of IEEE Computer Vision and Pattern Recognition*, pages 3020–3028, 2017.

[15] Ryosuke Tanno, Daichi Horita, Wataru Shimoda, and Keiji Yanai. Magical rice bowl: A real-time food category changer. In *Proc.of ACM International Conference Multimedia*, pages 1244–1246, 2018.

[16] Parneet Kaur, Karan Sikka, Weijun Wang, Serge Belongie, and Ajay Divakaran. Foodx-251: A dataset for fine-grained food classification. In *arXiv preprint arXiv:1907.06167*, 2019.

[17] Weiqing Min, Linhu Liu, Zhiling Wang, Zhengdong Luo, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network. In *Proc.of ACM International Conference Multimedia*, pages 393–401, 2020.

[18] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. In *arXiv preprint arXiv:2109.01134*, 2021.

[19] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Proc.of Advances in Neural Information Processing Systems*, 29, 2016.

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Proc.of Advances in Neural Information Processing Systems*, 30, 2017.

[21] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[22] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc.of IEEE Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *arXiv preprint arXiv:1711.05101*, 2017.

[25] Kaimu Okamoto and Keiji Yanai. Uec-foodpix complete: A large-scale food image segmentation dataset. In *Proc.of International Conference on Pattern Recognition*, pages 647–659. Springer, 2021.

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Proc.of Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.