Simultaneous Multiple-Prompt Guided Generation Using Differentiable Optimal Transport

Yingtao Tian  
Google Brain  
Tokyo, Japan  
alantian@google.com

Marco Cuturi  
Google Brain (currently at Apple)  
Paris, France  
cuturi@apple.com

David Ha  
Google Brain  
Tokyo, Japan  
hadavid@google.com

Abstract
Recent advances in deep learning, such as powerful generative models and joint text-image embeddings, have provided the computational creativity community with new tools, opening new perspectives for artistic pursuits. Text-to-image synthesis approaches that operate by generating images from text cues provide a case in point. These images are generated with a latent vector that is progressively refined to agree with text cues. To do so, patches are sampled within the generated image, and compared with the text prompts in the common text-image embedding space; The latent vector is then updated, using gradient descent, to reduce the mean (average) distance between these patches and text cues. While this approach provides artists with ample freedom to customize the overall appearance of images, through their choice in generative models, the reliance on a simple criterion (mean of distances) often causes mode collapse: The entire image is drawn to the average of all text cues, thereby losing their diversity. To address this issue, we propose using matching techniques found in the optimal transport (OT) literature, resulting in images that are able to reflect faithfully a wide diversity of prompts. We provide numerous illustrations showing that OT avoids some of the pitfalls arising from estimating vectors with mean distances, and demonstrate the capacity of our proposed method to perform better in experiments, qualitatively and quantitatively.

Introduction

The computational creativity community has been at the forefront of engaging with recent advances in deep learning, adopting early on generative models that are able to produce high-quality text and images. Such models offer varying degrees of realism and control to the artist, enabling the generation of results with artistic value. Recent advances have brought forward models that can produce images from natural language prompts, using pre-trained image generative models guided by text descriptions ([Radford et al. 2021]). The computational creativity community has seized this opportunity, has shared large bodies of code ([Burton-King 2021]; [Murdock 2021a]) and generated a large body of artwork, some of which has been curated online ([Snell 2021]; [Murdock ]).

These tools are favored by artists because they can shape generation in various ways: For instance, a relevant genera-

(a) Generated images from two-prompts using our method. (Left) “Walt Disney World.” and “a daytime picture of Tokyo.” (Right) “A painting of cat.” and “A painting of dog.”.

(b) The architecture of our work. Iteratively, the loss is computed forward (marked by →) and the gradient is calculated backward (marked by ←) to update the latent variable z.

Figure 1: Our method illustrated with generated images and the architecture. In contrast, the existing method would fail with these two-prompts, producing images with less diverse features (left) or a painting with much different art style than single prompt (right). This is because the existing method of taking the mean cannot treat different parts of the image separately, and vector arithmetic in the latent space introduces uncontrollable changes in the semantics. Detailed analysis can be found in text. All figures in this paper are generated using pre-trained CLIP and VQGAN models, both publicly released under MIT license.
Prompt Guided Image Generation

A notable trend in the field of computational creativity is to guide image generation using natural language as prompts. These text-to-painting synthesis tools allow artists to specify the content of a painting using prompts from natural languages. This text-driven generation has revolutionized the computational generation of artworks, as evidenced in online curated collections [Snell 2021][Murdoch]. These advances are made possible by combining two innovations from deep learning:

- Powerful image generative models. Such models include recent generative adversarial networks (GANs) [Karras, Laine, and Aila 2019][Karras et al. 2020][Karras et al. 2021], variational autoencoders [van den Oord, Vinyals, and Kavukcuoglu 2017] and diffusion models [Ho, Jain, and Abbeel 2020][Song, Meng, and Ermon 2020][Nichol and Dhariwal 2021][Dhariwal and Nichol 2021], that can produce images with high fidelity and diversity. Formally, this process can be denoted as $x = G(z)$ where the generative model $G : \mathbb{R}^d \rightarrow \mathbb{R}^{h \times w \times 3}$ converts a latent space variable $z \in \mathbb{R}^d$ to an RGB image of height $h$, weight $w$ and 3 color channels. $x \in \mathbb{R}^{h \times w \times 3}$. $z$ could be further manipulated to allow generating more suitable $x$ [Li, Jin, and Zhu 2021], allowing artist to control the generation of artworks that fall in desired genres [Jin et al. 2017].

- Joint modeling of images and natural language. This idea has been long in the making [Thomee et al. 2016][Li et al. 2017], but only recently given a convincing implementation thanks to progress in natural languages modeling [Raffel et al. 2019][Brown et al. 2020], and notably the ability to embed jointly images and text so well that the need for task-specific fine-tuning is eliminated, as shown in CLIP [Radford et al. 2021]. CLIP provides two jointly-trained, differentiable encoders, $E_I : \mathbb{R}^{h \times w \times 3} \rightarrow \mathbb{R}^d$ and $E_T : T \rightarrow \mathbb{R}^d$, for image and text respectively. We do not further elaborate the domain of $T$ as it is not the focus of this work. Formally, given an image $x$ and a text $t$, and a distance function $D : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+$ the encoded image $u = E_I(x)$ and the encoded text $v = E_T(t)$ are in a common comparable space $U = \mathbb{R}^d$, and $D(u, v)$ measures the similarity between $x$ and $t$. In the
case of CLIP that is trained with cosine distance, practically $D$ could be chosen as cosine distance or geodesic distance, both effectively measuring the angle between the two vectors and being trivially differentiable. Ideally, text-driven image generation is now feasible by iteratively adjusting the latent space vector $z$, to minimize $D(u, v)$, the distance between the encoded image $x = G(z)$ and encoded user-specific prompt $t$. As $G$, $E_l$ and $D$ are differentiable, $z$ could be updated using gradient descent: $z ← z - γ∇_z F(z)$ where $∇_z F$ is the gradient of $F$ defined as $F(z) = D(E_l(G(z)), E_l(t))$ and $γ$ is a learning rate.

Using a distance from a single image to a single prompt is usually too restrictive. Therefore, and in practice, the distance is computed over pairs of multiple images and texts as follows: On the image side, $n$ patches (a.k.a. cutouts. We use these two terms interchangeably), which we denote as $x_1, \cdots, x_n = S(x)$ are randomly sampled from image $x$ in the fashion of image data augmentation (Shorten and Khoshgoftaar 2019). We assume $x_i \in \mathbb{R}^h \times \mathbb{R}^w$ still holds since we can trivially add a resizing step at the end of augmentation. This practice serves as a regularizer to ensure numerical stability and avoid fitting into regions of $z$ where $G$ has bad support. On the text side, $m$ text prompts, denoted as $t_1, \cdots, t_m$, are often considered, which allows artists to explore the possibilities of art by combining multiple texts as directions. Again, they are encoded accordingly, giving $u_1, \cdots, u_n : u_i = E_l(x_i) \in \mathbb{R}^d$ and $v_1, \cdots, v_m : v_j = E_l(t_j) \in \mathbb{R}^d$. These pairwise distances are then combined to form a loss, which is

$$F(z) = \text{Mean}_D(z) \overset{\text{def}}{=} \frac{1}{mn} \sum_{1 \leq i \leq n, 1 \leq j \leq m} D(u_i, v_j), \quad (1)$$

and thus the gradient $∇_z F$ reads

$$∇_z F = \left( \sum_{1 \leq i \leq n} \frac{∂\text{Mean}_D}{∂u_i} \frac{∂u_i}{∂x_i} \frac{∂x_i}{∂z} \right) \frac{∂x}{∂z} \quad (2)$$

where

$$\frac{∂\text{Mean}_D}{∂u_i} = \frac{1}{nm} \sum_{1 \leq j \leq m} \frac{∂D(u_i, v_j)}{∂u_i}$$

$$\frac{∂u_i}{∂x_i} = \nabla_x E_l(x_i), \quad \frac{∂x_i}{∂z} = \nabla_z G(z) \quad (3)$$

and $∂x_i/∂x$ is defined as long as the random sampling is differentiable w.r.t. the input image $x$ which is often the case of data augmentations. This framing of text-driven generation has been applied to different generators $G$, yielding a variety of artistic results: using unconditional GAN generation, like BigGAN (Murdock 2021a), VQGAN (Burton-King 2021) and SIREN (Murdock 2021b), conditional generation using GAN, such as StyleCLIP (Patashnik et al. 2021), that enables editing existing images. In addition to GANs, it can also be applied to Diffusion models (Crowson 2021) [Kim and Ye 2021] [Nichol et al. 2021].

Differentiable Optimal Transport

Optimal transport (OT), as its name suggests, can be understood as finding an efficient way to ‘move’ or ‘transport’, the mass from a probability distribution to another distribution. We borrow notations from the survey book (Peypé and Cuturi 2021; Nichol et al. 2021). Differentiable Optimal Transport (Singer et al. 2020) and focus on one of the canonical OT formulations, one that was proposed in (Kantorovich 1942). A discrete measure with weights $a$ on locations $u_1, \cdots, u_n$, would be denoted as $α = \sum_{1 \leq i \leq n} a_i δ_{u_i}$, where notation $δ_{u_i}$ stands for a Dirac mass at location $u_i$. Similarly, for weights $b$ on locations $v_1, \cdots, v_m$, we have $β = \sum_{1 \leq j \leq m} b_j δ_{v_j}$. A possible way to map a discrete measure $α$ onto $β$, given a cost matrix $C \in \mathbb{R}^n \times m$, can be represented with a coupling matrix $P \in \mathbb{R}^n \times m$, where the amount of mass transported from the $i$-th location in $α$ to $j$-th location in $β$ is stored as $P_{i,j}$. The set of admissible couplings, $U$, is defined through $a$ and $b$ as

$$U(a, b) \overset{\text{def}}{=} \left\{ P \in \mathbb{R}^n \times m : \sum_j P_{i,j} = a_i, \sum_i P_{i,j} = b_j \right\} .$$

These row- and column-sum constraints for $P$ indicate that the entire mass from $α$ is indeed transported to $β$. Kantorovich’s problem of interest is

$$L(α, β, C) \overset{\text{def}}{=} \min_{P \in U(a, b)} (C, P) \overset{\text{def}}{=} \sum_{i,j} C_{i,j} P_{i,j}$$

which can be solved using linear programming, notably network flow solvers. The linear programming route, while well established, has a few drawbacks: it is slow, with an unstable solution. A possible workaround is to add an entropic regularization term, where the entropy of $P$ reads

$$H(P) \overset{\text{def}}{=} -\sum_{i,j} (P_{i,j} \log(P_{i,j}) - 1) .$$

The regularized problem reads:

$$L^\varepsilon(α, β, C) \overset{\text{def}}{=} \min_{P \in U(a, b)} (C, P) - \varepsilon H(P) .$$

This regularization has several practical virtues: the regularized problem can be solved efficiently with Sinkhorn’s Algorithm (Cuturi 2013; Séjourné et al. 2019), a fast iterative algorithm that only uses matrix-vector arithmetic. Another advantage, equally important in our setting, is that this approach, as implemented in OTT-JAX (Cuturi et al. 2022), results in fully differentiable quantities. Namely, assume that the cost matrix $C$ is provided in the form of a differentiable function resulting in entries $C_{i,j} \overset{\text{def}}{=} C(u_i, v_j)$. Then the gradient of $L^\varepsilon$ w.r.t. $u_i$ exists and is defined everywhere:

$$\forall i, 1 \leq i \leq n, \left| \frac{∂L^\varepsilon}{∂u_i} \right| < \infty .$$

Not that the optimal solution $P^\varepsilon$ corresponding to $L^\varepsilon$ can also be differentiated w.r.t. any of the relevant inputs, using the implicit function theorem (Krantz and Parks 2002), as proposed in OTT-JAX (Cuturi et al. 2022), but this is not used in this paper because we rely on Danskin’s Theorem (Dansk 1966) (a.k.a Envelope Theorem) to differentiate $L^\varepsilon$ w.r.t. $C$.

Methodology

Our motivation comes from the concern arising from using an averaged loss $\text{Mean}_D$. By focusing on means, all sampled
patches are encouraged to move uniformly to the mean of all prompts. This undermines the very motivation of introducing multiple prompts, which is to allow artists to obtain spatial diversity in the generated images, with various areas reflecting the diversity prompts. Furthermore, taking the mean in the embedding space introduces gradients in unwanted directions. Since the locations in the embedding space are associated with semantics, doing so may introduce uncontrollable, redundant semantics. To make things worse, the mean arithmetic effectively assumes an Euclidean space, which is inconsistent to the CLIP model that is trained with cosine distance in the embedding space.

To address these issues, it is possible to devise an arithmetic in non-Euclidean Space. However, finding a proper choice that works well with the rest of pipeline is not trivial and warrants a separate study. Instead we propose to eliminate the undesired simplifications brought by mean arithmetic, to replace Mean in Equation 1 with an optimal transport loss,

$$\mathcal{F} = \mathcal{F}_\epsilon (a, b, \{D(u_i, v_j)\})_{i,j}$$  \hspace{1cm} (5)$$

where \( a_i = 1/n \) and \( b_j = 1/m \), and the cost matrix \( \mathbf{C} \) is populated with pairwise distance \( D \) evaluations. Now, the gradient \( \nabla_z \mathcal{F} \) reads

$$\nabla_z \mathcal{F} = \left( \sum_{1 \leq i \leq n} \frac{\partial \mathcal{F}_\epsilon}{\partial u_i} \frac{\partial x_i}{\partial x} \right) \frac{\partial x}{\partial z}. \hspace{1cm} (6)$$

Comparing with Equation 2, the only different term is \( \frac{\partial \mathcal{F}_\epsilon}{\partial u_i} \), which is also defined as in Equation 4. Along with other terms (see Equation 3), all terms are defined, and thus we know that \( \nabla_z \mathcal{F} \) is also well-defined and can be used in the iteratively updating of \( z \):

$$z \leftarrow z - \gamma \nabla_z \mathcal{F}$$

In doing so, the above mentioned issues are solved for the following reasons:

**OT Treats different patches differently.** As OT matches patches and text prompts, it naturally introduces a distinct treatment of patches according to their distances to text prompts. As the patches are randomly sampled, it encourages the intrinsic diversity inside a single generated image.

**OT does not involve arithmetic in the latent space.** OT relies on distances, but does not use averages in embedding spaces. Therefore it does not produce synthetic prompts in embeddings space that may not correspond to semantics. Furthermore, OT is agnostic to how distances are defined: any distance, other than cosine distance or geodesic distance, could be used to populate matrix \( \mathbf{C} \).

**Experiments**

In this section, we highlight a few possibilities brought forward by using our methodology when handling multiple text prompts. Due to the creative nature of text-to-image synthesis, there is no standard measuring stick, such as classification accuracy, to provide a simple comparison between methods. Nevertheless, we consider a few tasks that can help us gain insight into the novelty, the properties and the behavior of our method. We consider:

- **Generated Image.** Naturally the foremost task is to show the generated image \( x \) with multiple text prompts \( t_1, \ldots, t_m \). In this task, we focus on whether the generated image represents the text prompts in a way that is distinctive and subjectively recognized by human viewers.

- **Patches (Cutouts) from Generated Images.** Our method improves the diversity of patches through increasing the correlation between the distribution of randomly sampled patches and multiple text prompts, as we identify as a source of issues from existing practices. In this task, we show the patches and organize them by text prompt. Formally, we show the \( n \) patches \( x_1, \ldots, x_n \) sampled from \( x \), and group \( x_i \) by \( j^* = \arg \min_j D(u_i, v_j) \), the closest text prompt in
Figure 3: The cutouts (patches) from generated images in Figure 2 for both our proposed method (OT) and the baseline. We show in (a) and (b) the sampled patches. Then in (c) and (d) we group these patches by the closer (measure by $D$) prompt they are to. Due to space constraints, we only show the number of each group and six patches that are mostly closet.

**Tangent of Patches (Cutouts) on Cost Plane.** We identify the issue materialize in the way gradient information is pass from $F$ back to patches, which is $\partial \text{Mean}_D / \partial u_i$ part in Equation 2 and propose to use $L_C^*$ such that the $\partial L_C^* / \partial u_i$ part in Equation 6 is better.

To quantitatively qualify such property, a few extra deliberations are needed. Concretely, we first define

$$\phi(u_i) : \mathbb{R}^d \rightarrow \mathbb{R}^m \equiv [D(u_i, v_1), \cdots, D(u_i, v_m)],$$

which is by definition a differentiable mapping from the aforementioned embedding space $\mathbb{R}^d$ to $\mathbb{R}^m$, a $m$-d space of distances to prompts where the $j$-th element is the distance to prompt $j$. As $\partial L_C^* / \partial u_i \in T_{u_i}$ (the tangent space of $\mathbb{R}^d$ at $u_i$), the pushforward by $\phi$ at $u_i$ is defined as $d\phi : T_{u_i} \mathbb{R}^d \rightarrow T_{\phi(u_i)} \mathbb{R}^m$ such that when applied to the gradient,

$$w_i = d\phi(\partial L_C^* / \partial u_i)$$

is in the tangent space of $\mathbb{R}^m$. Intuitively, $w_i$ is a $m$-dimensional vector whose $j$-th element denotes the component of gradient that moves the $i$-th patch towards the $j$-th text prompt.

**Comparing our Method with the Baseline for Two Prompts Setting**

In this experiment, we focus on a scenario with $M = 2$ prompts, “Walt Disney World.” and “daytime picture of Tokyo.”
Tokyo.” We compare two models, our proposed approach with Optimal Transport (Equation 5) and the baseline using Mean (Equation 1), with the purpose of investigating the behavior of these methods and the difference made by our approach. We keep all other configurations the same. Namely, we use a pre-trained VQGAN [Esser, Rombach, and Ommer 2021] on ImageNet dataset, \( N = 64 \) randomly sampled patch, and 1000 iterations of updating \( z \). We organize the conducted tasks as explained before.

**Generated Image and Patches (Cutouts) from it.** In Figure 2 we show the generated image from both methods. Also in Figure 3 we show the patches (cutouts) sampled from the generated images at the end of all iterations.

We observe that OT helps generate images where patches (cutouts) are more balanced (36/28 vs 49/15). Furthermore, OT’s results are more diverse for two prompts. For OT, patches close to “Walt Disney World.” are more like close-ups and patches close to “A daytime picture of Tokyo.” are mostly zoomed-out. As patches are randomly done, it reflects the intrinsic property of generated images.

**Our Method’s Behavior with Multiple Prompts**

Having comparing our OT-based method with the baseline on the two prompts setting, we shift our focus to the scenario where our method is applied to multiple prompts. As this is we designed our method to expose fine differentiation among prompts, it becomes interesting to investigate such behavior when the number of prompts increases. In doing so, we consider totally \( M = 6 \) prompts, numbered from \( P_0 \) to \( P_5 \):

- **\( P_0 \): Impressionism / Edgar Degas/ Landscape at Valery-sur-Somme**
- **\( P_1 \): Impressionism Laszlo Mednyanszky/ Landscape in the Alps (View from the Rax)**
- **\( P_2 \): Romanticism / J.M.W. Turner/ The Lake, Petworth, Sunset; Sample Study**
- **\( P_3 \): Romanticism / George Stubbs/ Hound Coursing a Stag**
- **\( P_4 \): Realism / Alexey Venetsianov/ In the Fields. Spring**
- **\( P_5 \): Realism / Alexey Venetsianov/ A Peasant Woman with Scythe and Rake**

and as the prompts suggest, we use a pre-trained VQGAN on WikiArt dataset consisting mostly of paintings. The purpose is to both show that our method could be applied to generative models trained from different genre data, and also that the painting allows easier qualitative comparison of both objects and artistic styles. As the same setting mentioned above, \( N = 64 \) randomly sampled patch, and 1000 iterations of updating are used. We conduct tasks as explained before.

**Generated Image.** In Figure 5 we show in the first group the generated images corresponding to these prompts individually, and in the second group the generated images by combining prompts using our proposed method. We observe that our method is capable of composing the instructions from several prompts, in terms of styles and objects, into the same canvas.
Figure 5: The generated images from multiple (6) prompts, labeled P0 to P5. (a) - (f): The first group of 6 images are generated using each one prompt respectively, as a controlling group. (g) - (i): The second group of 6 images are the generated images with multiple (1 to 6) prompts respectively from our proposed method, each one of which using a combination of multiple problems specified in the caption.
Figure 6: Tangent, representing the gradients on patches (cutouts) after they are pushed forward to Cost Plan. The first group is for the generation with 3 prompts and the second group is for the generation with 6 prompts, showing in 2D and 3D slices.

Tangent of Patches (Cutouts) on Cost Plane. In Figure 6, we show that the good behavior on tangent remains even for multiple prompts. This means that our method is capable of guiding generating images that are diverse in its contents w.r.t. multiple prompts.

Conclusion and Future Work

In this paper we discuss the problem in dealing with multiple text prompts in the setting of text-driven image generation for computational creativity setting. We then propose to address the issue using OT (Optimal Transport) between sampled patches in the generated image and multiple text prompts, and show its theoretical motivation and quantitative and qualitative empirical results highlighting the advantage brought by our proposed method.

One of the advantages in our method is that it is in theory orthogonal to other parts in the whole text driven image generation pipeline, as we show primarily that it works for VQGAN trained on several datasets. We envision that future work would investigate leveraging our proposed method to other drastically different forms of generative method, such as diffusion models. Another possible future direction may principally study the combination of optimal transport and adaptive sampling where in our proposed work only random sampling is used for simplicity.

Author Contributions

Yingtao: Ideated the problem and the method, conducted experiments, drafted the paper.
Marco Cuturi: Provided advises, helped designing the method / experiments, helped drafting the paper.
David Ha: Provided advises and gave feedback / suggestions for the whole work, helped drafting the paper.

Acknowledgement

The authors thank Yujin Tang, Jerry Li, Mathieu Blondel for their valuable comments and suggestions.

References

[Brown et al. 2020] Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.
[Burton-King 2021] Burton-King, S. 2021. Introduction to vqgan+clip. https://bit.ly/3rcedh4.
