Appendix for "Cross-Ray Neural Radiance Fields for Novel-view Synthesis from Unconstrained Image Collections"

In the appendix, we provide detailed proofs of the proposition, more details, and more experimental results of the proposed Corss-Ray Neural Radiance Fields (CR-NeRF)[1]. We organize the appendix into the following sections.

- In Sec. A we provides proof of our Proposition 1.
- In Sec. B, we provide details on inference of our CR-NeRF.
- In Sec. C, we discuss the impact of the number of rays on our CR-NeRF, which supports the necessity of considering multiple rays.
- In Sec. D we describe the detail of our grid sampling strategy.
- In Sec. E we qualitatively and quantitatively compare the transient network of our CR-NeRF with existing methods.
- In Sec. F we compare the training time of our CR-NeRF with existing methods.
- In Sec. G we discuss the effectiveness of our cross-ray paradigm and fusing level.
- In Sec. H we demonstrate more synthesized views by interpolating between an appearance embedding to another.
- In Sec. I, we report more qualitative experimental results of appearance modeling by comparing CR-NeRF and existing methods on Brandenburg Gate and Trevi Fountain datasets.
- In Sec. J we demonstrate more synthesized views by transferring appearance from unseen images.

A. Proof of Proposition 1

Proposition 1. *Given an invertible constant matrix* $\mathbf{P}\in\mathbb{R}^{C\times C}$*, assuming that* $\mathcal{F}^a\sim\mathcal{N}(\boldsymbol{\mu}_a,\boldsymbol{\Sigma}_a)$*,* $\mathcal{F}^{\mathrm{cr}}\sim\mathcal{N}(\boldsymbol{\mu}_{\mathrm{cr}},\boldsymbol{\Sigma}_{\mathrm{cr}})$ *and* $\mathcal{T}(\mathcal{F}^{\mathrm{cr}})\sim\mathcal{N}(\boldsymbol{\mu}_a,\boldsymbol{\Sigma}_a)$*, where* $\mathcal{T}(\mathcal{F}^{\mathrm{cr}})=\mathbf{T}(\mathcal{F}^{\mathrm{cr}}-\boldsymbol{\mu}_{\mathrm{cr}})+\boldsymbol{\mu}_a$ *and* $\mathbf{T}\in\mathbb{R}^{C\times C}$ *is a transformation matrix, the optimal* $\mathbf{T}$ *to Problem (5) is:*

$$\mathbf{T}=\boldsymbol{\Sigma}_{\mathrm{cr}}^{-1/2}\left(\boldsymbol{\Sigma}_{\mathrm{cr}}^{1/2}\mathbf{P}\boldsymbol{\Sigma}_a\mathbf{P}^\top\boldsymbol{\Sigma}_{\mathrm{cr}}^{1/2}\right)^{1/2}\boldsymbol{\Sigma}_{\mathrm{cr}}^{-1/2}\mathbf{P}^{-1}. \tag{A.1}$$

Proof. We rewrite Eqn. (5) using $\mathcal{T}(\mathcal{F}^{\mathrm{cr}})=\mathbf{T}(\mathcal{F}^{\mathrm{cr}}-\boldsymbol{\mu}_{cs})+\boldsymbol{\mu}_a$ as:

$$\mathbb{E}_{\mathcal{F}^{\mathrm{cr}},\mathcal{F}^a}[\mathbf{T}(\mathcal{F}^{\mathrm{cr}}-\boldsymbol{\mu}_{\mathrm{cr}})+\boldsymbol{\mu}_a-\mathcal{F}^a]^\top[\mathbf{T}(\mathcal{F}^{\mathrm{cr}}-\boldsymbol{\mu}_{\mathrm{cr}})+\boldsymbol{\mu}_a-\mathcal{F}^a]+\beta\left\{\mathbf{P}[\mathbf{T}(\mathcal{F}^{\mathrm{cr}}-\boldsymbol{\mu}_{\mathrm{cr}})+\boldsymbol{\mu}_a]-\mathcal{F}^{\mathrm{cr}}\right\}^\top\left\{\mathbf{P}[\mathbf{T}(\mathcal{F}^{\mathrm{cr}}-\boldsymbol{\mu}_{\mathrm{cr}})+\boldsymbol{\mu}_a]-\mathcal{F}^{\mathrm{cr}}\right\}. \tag{A.2}$$

Let $\boldsymbol{u}=\mathcal{F}^{\mathrm{cr}}-\boldsymbol{\mu}_{\mathrm{cr}}$, $\boldsymbol{v}=\mathbf{T}\boldsymbol{u}$, $\boldsymbol{w}=\boldsymbol{\mu}_a-\mathcal{F}^a$ and $\boldsymbol{\mu}_\Delta=\mathbf{P}\boldsymbol{\mu}_a-\boldsymbol{\mu}_{\mathrm{cr}}$, based on $\mathcal{F}^{\mathrm{cr}}\sim\mathcal{N}(\boldsymbol{\mu}_{\mathrm{cr}},\boldsymbol{\Sigma}_{\mathrm{cr}})$, $\mathcal{T}(\mathcal{F}^{\mathrm{cr}})\sim\mathcal{N}(\boldsymbol{\mu}_a,\boldsymbol{\Sigma}_a)$ and $\mathcal{F}^a\sim\mathcal{N}(\boldsymbol{\mu}_a,\boldsymbol{\Sigma}_a)$, we have $\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_{\mathrm{cr}})$, $\boldsymbol{v}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_a)$ and $\boldsymbol{w}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_a)$. Then Eqn. (A.2) can be rewritten as:

$$\mathbb{E}_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}}\left[\boldsymbol{v}+\boldsymbol{w}\right]^\top\left[\boldsymbol{v}+\boldsymbol{w}\right]+\beta\left[\mathbf{P}\boldsymbol{v}+\boldsymbol{\mu}_\Delta-\boldsymbol{u}\right]^\top\left[\mathbf{P}\boldsymbol{v}+\boldsymbol{\mu}_\Delta-\boldsymbol{u}\right]. \tag{A.3}$$

Let $\boldsymbol{v}^*=\mathbf{P}\boldsymbol{v}$, we obtain $\boldsymbol{v}^*\sim\mathcal{N}(\mathbf{0},\mathbf{P}\boldsymbol{\Sigma}_a\mathbf{P}^\top)$. Expanding Eqn. (A.3) to:

$$\mathbb{E}_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}}\left[\boldsymbol{v}^\top\boldsymbol{v}+\boldsymbol{v}^\top\boldsymbol{w}+\boldsymbol{w}^\top\boldsymbol{v}+\boldsymbol{w}^\top\boldsymbol{w}\right]+\beta\left[\boldsymbol{v}^{*\top}\boldsymbol{v}^*+\boldsymbol{\mu}_\Delta^\top\boldsymbol{\mu}_\Delta+\boldsymbol{u}^\top\boldsymbol{u}+\boldsymbol{v}^{*\top}\boldsymbol{\mu}_\Delta+\boldsymbol{\mu}_\Delta^\top\boldsymbol{v}^*-\boldsymbol{v}^{*\top}\boldsymbol{u}-\boldsymbol{u}^\top\boldsymbol{v}^*-\boldsymbol{\mu}_\Delta^\top\boldsymbol{u}-\boldsymbol{u}^\top\boldsymbol{\mu}_\Delta\right]. \tag{A.4}$$

Since $\boldsymbol{\mu}_\Delta$ is a constant, $\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_{\mathrm{cr}})$, $\boldsymbol{v}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_a)$ and $\boldsymbol{v}^*\sim\mathcal{N}(\mathbf{0},\mathbf{P}\boldsymbol{\Sigma}_a\mathbf{P}^\top)$, we obtain $\mathbb{E}_{\boldsymbol{v},\boldsymbol{w}}[\boldsymbol{v}^\top\boldsymbol{w}]=\mathbb{E}_{\boldsymbol{v},\boldsymbol{w}}[\boldsymbol{w}^\top\boldsymbol{v}]=0$, $\mathbb{E}_{\boldsymbol{v}}[\boldsymbol{v}^{*\top}\boldsymbol{\mu}_\Delta]=\mathbb{E}_{\boldsymbol{v}}[\boldsymbol{\mu}_\Delta^\top\boldsymbol{v}^*]=0$ and $\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}^\top\boldsymbol{\mu}_\Delta]=\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{\mu}_\Delta^\top\boldsymbol{u}]=0$. Then, Eqn. (A.4) can be represented as:

$$\mathbb{E}_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}}\left[\boldsymbol{v}^\top\boldsymbol{v}+\boldsymbol{w}^\top\boldsymbol{w}\right]+\beta\left[\boldsymbol{v}^{*\top}\boldsymbol{v}^*+\boldsymbol{\mu}_\Delta^\top\boldsymbol{\mu}_\Delta+\boldsymbol{u}^\top\boldsymbol{u}-2\boldsymbol{v}^{*\top}\boldsymbol{u}\right]. \tag{A.5}$$

[1]We suggest checking the video demo synthesized by our CR-NeRF in the supplementary.

According to the property of trace of matrix, minimizing Eqn. (A.5) is equivalent to minimizing:

$$tr\left(\mathbb{E}_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}}\left[\boldsymbol{v}\boldsymbol{v}^\top+\boldsymbol{w}\boldsymbol{w}^\top\right]+\beta\left[\boldsymbol{v}^*\boldsymbol{v}^{*\top}+\boldsymbol{u}\boldsymbol{u}^\top-2\boldsymbol{v}^*\boldsymbol{u}^\top\right]\right) \tag{A.6}$$

$$=tr\left(2\boldsymbol{\Sigma}_a+\beta(\boldsymbol{\Sigma}_a+\boldsymbol{\Sigma}_{\mathrm{cr}}-2\mathbb{E}_{\boldsymbol{u},\boldsymbol{v}}[\boldsymbol{v}^*\boldsymbol{u}^\top])\right) \tag{A.7}$$

Let $\Phi=\mathbb{E}_{\boldsymbol{u},\boldsymbol{v}}[\boldsymbol{v}^*\boldsymbol{u}^\top]$ denote the covariance of $\boldsymbol{v}^*$ and $\boldsymbol{u}$. Then, the optimal T to Equation 6 can be reformulated as:

$$\mathbf{T}=\arg\max_{\mathbf{T}}(tr(\Phi)). \tag{A.8}$$

Olkin et al. [34] show a unique solution to Eqn. (A.8) is

$$\Phi=\mathbf{P}\boldsymbol{\Sigma}_a\mathbf{P}^\top\boldsymbol{\Sigma}_{\mathrm{cr}}^{1/2}\left(\boldsymbol{\Sigma}_{\mathrm{cr}}^{1/2}\mathbf{P}\boldsymbol{\Sigma}_a\mathbf{P}^\top\boldsymbol{\Sigma}_{\mathrm{cr}}^{1/2}\right)^{-1/2}\boldsymbol{\Sigma}_{\mathrm{cr}}^{1/2}. \tag{A.9}$$

Since $\Phi=\mathbb{E}_{\boldsymbol{u},\boldsymbol{v}}[\boldsymbol{v}^*\boldsymbol{u}^\top]=\mathbb{E}_{\boldsymbol{u},\boldsymbol{v}}[\boldsymbol{v}^*(\mathbf{T}^{-1}\boldsymbol{v})^\top]=\mathbf{P}\mathbb{E}_{\boldsymbol{u},\boldsymbol{v}}[\boldsymbol{v}\boldsymbol{v}^\top](\mathbf{T}^{-1})^\top=\mathbf{P}\boldsymbol{\Sigma}_a(\mathbf{T}^{-1})^\top$, combining Eqn. (A.9) obtains the final $\mathbf{T}$

$$\mathbf{T}=\mathbf{P}^{-1}\boldsymbol{\Sigma}_{\mathrm{cr}}^{-1/2}\left(\boldsymbol{\Sigma}_{\mathrm{cr}}^{1/2}\mathbf{P}^\top\boldsymbol{\Sigma}_a\mathbf{P}\boldsymbol{\Sigma}_{\mathrm{cr}}^{1/2}\right)^{1/2}\boldsymbol{\Sigma}_{\mathrm{cr}}^{-1/2}. \tag{A.10}$$

□

B. Inference of CR-NeRF

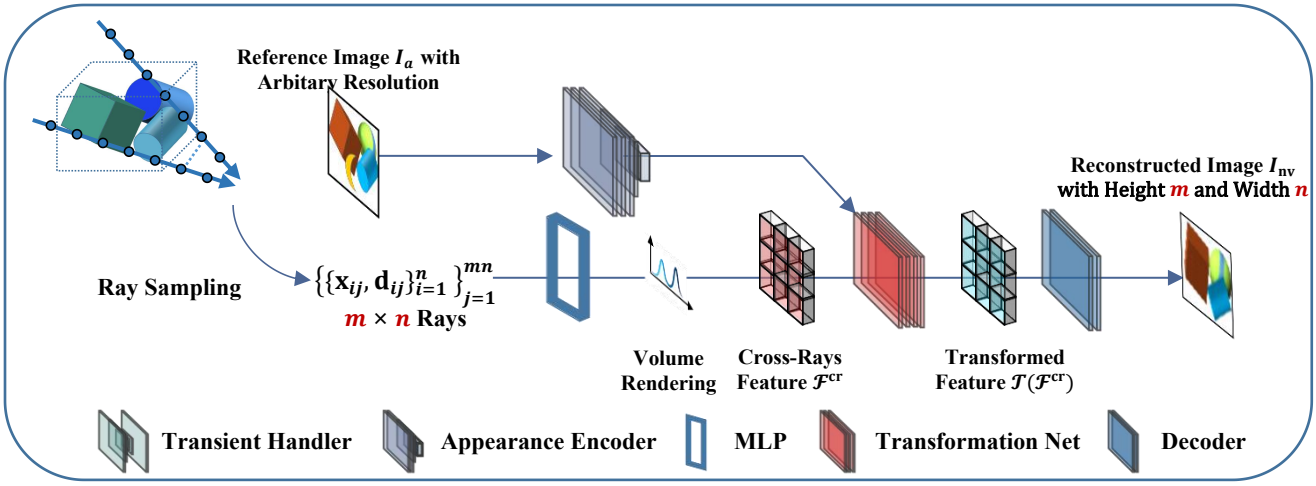

Figure 7. Illustration on inference of CR-NeRF.

We provide details on inference of our CR-NeRF in Fig. 7 and Alg. 2. During inference, we sample $m \times n$ rays, which intersect with $m \times n$ pixels of reconstructed image I_{n} (m and n equals the height and width of I_{n}). Thanks to our encoder parameterized by convolutional neural network and adaptive average pooling, the reference image can be of arbitrary size and we generate an appearance embedding $\mathcal{F}^{\mathrm{a}}$ by encoding I_{n} with appearance encoder. After representing the $m \times n$ rays with our proposed cross-ray feature $\mathcal{F}^{\mathrm{cr}}$, we fuse $\mathcal{F}^{\mathrm{cr}}$ and $\mathcal{F}^{\mathrm{a}}$ with a transformation net and then decode the fused feature to synthesize I_{n}. During inference, we discard the transient object handler and content encoder.

Algorithm 2: The Inference pipeline of CR-NeRF.

Input: $m * n$ rays $\{\mathbf{r}_i\}_{i=1}^{m*n}$, a reference image $\mathcal{I}_a$ with size $m * n$, a multilayer perceptron MLP_{θ_1}, an appearance encoder E_{θ_2}, a transformation net $\mathcal{T}_{\theta_3}$, a decoder D_{θ_4}.

Output: The estimated colors of $m * n$ pixels of a novel view.

1 Generate cross-ray features $\mathcal{F}^{\mathrm{cr}}$ and appearance feature $\mathcal{F}^{a}$ with E_{θ_2} and MLP_{θ_1} by Eqn. (4).

2 Injecting appearance from $\mathcal{I}_a$ to scene representation by fusing $\mathcal{F}^{\mathrm{cr}}$ and $\mathcal{F}^{a}$ via $\mathcal{T}_{\theta_3}$.

3 Estimating color $\hat{\mathbf{c}}(\{\mathbf{r}_i\}_{i=1}^{m*n})$ w.r.t. the rays and the reference image by leveraging D_{θ_4}.

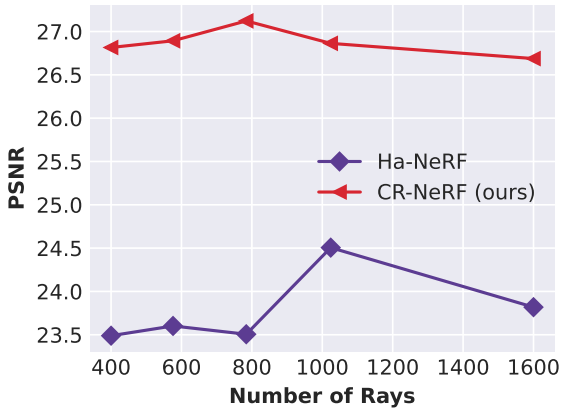

Figure 8. Effectiveness of different number of rays on our CR-NeRF and Ha-NeRF on brandenburg dataset in terms of PSNR.

#Rays	400	576	784	1024	1600
Ha-NeRF	23.49	23.60	23.51	24.51	23.82
CR-NeRF	26.82	26.89	27.12	26.86	26.69

Table 5. Effectiveness of different number of rays on our CR-NeRF and Ha-NeRF on brandenburg dataset in terms of PSNR.

C. Effectiveness of Number of Rays

We analyze the impact of the number of rays (#rays) on both Ha-NeRF and our (CR-NeRF). Fig. 8 shows the PSNR results of the two methods on the Brandenburg dataset in terms of different values of #rays. CR-NeRF consistently outperforms Ha-NeRF across all tested values of #rays, which verifies that considering multiple rays consistently boosts the performance of CR-NeRF. Additionally, the performance of CR-NeRF increases as the number of rays increases. However, we also note that after the number of rays exceeds 784, the performance of CR-NeRF starts to degrade gradually. One possible explanation is that increasing #rays over a threshold introduces ambiguity in view-consistent modeling, which harms the quality of synthesized views. Note that although Ha-NeRF uses multiple rays as input, information from each individual ray does not intersect with that of the others.

D. Grid Sampling strategy

Grid sampling strategy aims to extract a grid of $k \times k$ image pixels from a reference image, guided by a grid center $\mathbf{u}$ and sampling scale s. As detailed in [36] and illustrated in Fig. 9, GS involves uniformly selecting $k \times k$ image pixels based on the coordinate set $\mathcal{P}(\mathbf{u}, s) = \left\{(sx + u_x, sy + u_y) \mid x, y \in \left\{-\frac{k}{2}, \ldots, \frac{k}{2} - 1\right\}\right\}$, where $\mathbf{u}{=}(u_x, u_y){\in}\mathbb{R}^2$ and $s{\in}\mathbb{R}^+$. With these pixel coordinates, we sample $k \times k$ rays for cross-ray synthesis and also sample our predicted visibility mask for transient handling.

E. Comparisons of Transient Network

To further verify the effectiveness of our proposed transient network, we conduct a comparative analysis by replacing the transient network in our CR-NeRF with that of NeRF-W (termed CR-NeRF-U) and utilizing uncertainty formulation for training. The results in Tab. 6 show the superiority of our transient network on three datasets. Moreover, we visualize the output of the transient networks of CR-NeRF and NeRF-W in Fig. 10. CR-NeRF achieves a more accurate prediction by identifying the semantic feature of tourists and trees. Our transient network outperforms NeRF-W because predicting object visibility is much easier than predicting the colors and densities of transient objects.

F. Comparison of Training Time

In Tab. 7, we report the training times for CR-NeRF, Ha-NeRF, and NeRF-W, spanning 20 epochs, are 1583, 1701, and 1504 minutes, respectively. We employ 8 TITAN Xp GPUs with 17200 iterations per epoch.

G. Effectiveness of Cross-Ray Paradigm and Fusing Level

We study the effectiveness of our cross-ray paradigm and on which level to fuse with appearance features. To this end, we construct CR-NeRF-R, the only difference of CR-NeRF-R and CR-NeRF is that CR-NeRF-R conduct appearance transfer by considering *ray points of different rays* but CR-NeRF achieves the transfer on *different rays*. In other words, CR-NeRF-R fuses an *image-level* appearance feature $\mathcal{F}^a$ with *ray-point level* features, while CR-NeRF combines $\mathcal{F}^a$ and $\mathcal{F}^{\text{cr}}$. From Fig. 11, CR-NeRF is able to model a more accurate appearance, while also reconstructing a more consistent geometry. These results verify the superiority of the cross-ray manner and show fusing the image-level appearance features with cross-ray features is more effective than with the cross-ray-points features.

H. Interpolation of Appearance Embedding

Our proposed CR-NeRF is able to synthesize images that gradually change from one appearance image to another. We achieve this by linearly interpolating the appearance features of the two appearance images. From Fig. 12, we observe that (1) CR-NeRF is able to handle transient objects and thus synthesize non-transient images (*e.g.,* images in the second row of Fig. 12 have no transient objects, such as visitors in appearance 1, and the ground synthesized by CR-NeRF better shows the reflection effect of ground in appearance 1). (2) CR-NeRF captures the appearance more accurately than Ha-NeRF (*e.g.,* the sky color in the third row of is not as accurate as the fourth row of Fig. 12).

I. Modeling Appearance from Brandenburg and Trevi

We show qualitative experimental results of appearance modeling using images from Brandenburg and Trevi. As shown in Fig. 13 and Fig. 14, we transfer appearance from Brandenburg to Brandenburg and Trevi and vice versa. CR-NeRF recovers a more accurate appearance than Ha-NeRF, which demonstrates the effectiveness of our cross-ray paradigm.

J. Modeling Appearance from Unseen Images

Our proposed CR-NeRF is able to deal with unseen appearance images thanks to the ability of our cross-ray appearance modeling handler. As shown in Fig. 15, our CR-NeRF captures the whole range appearance (*e.g.,* the blue and purple appearance in the last two columns in Brandenburg and Trevi fountain datasets) of the reference image more accurately compared with Ha-NeRF. Moreover, our CR-NeRF synthesizes a more consistent appearance than images generated by Ha-NeRF (*e.g.,* the sudden bright light on the sky of the second and fourth column in the Brandenburg dataset). We also provide videos of unseen transfers on videos in the supplementary material. Note that NeRF-W needs to optimize its appearance embedding on each test image by pixel-level supervision, thus NeRF-W cannot be directly applied to unseen appearance transfer.

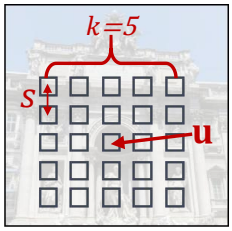

Figure 9. Illustration of grid sampling strategy.

Figure 10. Comparisons of the transient networks of CR-NeRF and NeRF-W.

	Brandenburg	Sacre	Trevi
CR-NeRF-U	24.72/0.8873	20.88/0.8161	20.84/0.7382
CR-NeRF	**26.86/0.9069**	**22.03/0.8369**	**22.02/0.7488**

Table 6. PSNR/SSIM of CR-NeRF-U on three datasets.

Method	EPOCH	Iteration	Time (minutes)
NeRF-W	20	17200	1504
Ha-NeRF	20	17200	1701
CR-NeRF	20	17200	1583

Table 7. Training time comparisons of different methods.

(a) Comparison of synthesized views **(b) Comparison of generated depth maps**

Figure 11. Comparison of CR-NeRF and CR-NeRF-R regarding detailed appearance and depth maps. CR-NeRF is able to synthesize a more accurate appearance (*e.g.,* the color of the statue in Trevi Fountain and in Sacre Coeur). Moreover, CR-NeRF successfully estimates the depth of the cavity portion of the building while CR-NeRF-R fails.

Figure 12. Interpolating between appearance 1 and appearance 2 with a fixed camera position (synthesized results are in the dashed box).

Figure 13. Transferring appearance from Brandenburg Gate to Brandenburg Gate and Trevi Fountain.

Figure 14. Transferring appearance from Trevi Fountain to Brandenburg Gate and Trevi Fountain.

Figure 15. Transferring appearance from unseen images to Brandenburg Gate and Trevi Fountain datasets.