Text-driven Video Manipulation with GPT4V

Tom James, John Feng, Willion Liang

1. Introduction

Generative Adversarial Networks (GANs) [11] have revolutionized image synthesis including faces and scenes, with recent style-based generative models [4, 8, 13, 14, 29]. Traditionally, researchers used the Gaussian distribution as the input latent code and then generate images. Recently, GAN inversion techniques are utilized to retrieval the latent code by the pretrained model from the real images. After that, the GAN generator can reconstruct the image using the estimated latent code, which provides challenges and opportunities for the image manipulation task. In our project, we work on semantic manipulation meaning changing the semantic attributes of an object.

Recent work [1, 2, 9, 12, 14] has two directions including optimizing the latent code and predicting the latent code via an image encoder directly. For image-level editing applications several approaches [1, 13] find specific semantic directions in the latent space, e.g., changing poses, colors, or age, while others [15] aim to change the global style, e.g., photo $\rightarrow$ skecth, photo $\rightarrow$ cartoon. More recently, StyleCLIP [29] was proposed to edit the image simply by words or phrases, like “angry”, “sad” and “getting older”. It takes advantage of CLIP [30], a vision and language model which was trained on 400 million real-world text-image pairs by using contrastive learning algorithm. With these image GAN inversion-based semantic editing approaches, how can we extend them to videos?

One straightforward method is to use state-of-art GAN inversion techniques and semantic editing like StyleCLIP to manipulate each frame in the video and then combine them to produce the video. However, the edited frames from a same video may not be consistent due to the lack of temporal constraint.

In this project, we start from the existing GAN inversion approaches to obtain the latent code frame by frame. Then we apply semantic editing like StyleCLIP to to edit the latent code for each frame. To deal with the limitation of lack of temporal constraint, we apply hierarchical pair-based optimization by minimizing the bi-directional photometric loss and the local generator regularization. In our experiments, we show that our new optimization approach helps achieve significantly improved temporal consistency while maintaining the editing style than baseline methods.

2. Related Work

Multimodal learning [19, 29, 30, 36] is the task of using machine learning algorithms to learn knowledge from different modalities, which received the huge attention these days. Language and vision is one of the most important directions for many tasks including image captioning [22–24], video captioning [10], language-based image or video retrieval [5], representation learning [19] and text-guide image generation [20, 25, 30].

Training representations from scratch cost the large amount of time and computing resource. Learning with the pretrained language or vision models become the new trend. Following the better performance of transformer architecture [35] in both vision and language tasks, a transformer-based pretrained language model, BERT [7] achieves success in various language tasks. As for the image representation learning, some popular CNN based pretrained models contains ResNet, VGGNet and AlexNet [32, 33]. While the Transformer architecture has become the standard for natural language processing tasks, its applications to computer vision remain limited. ViT [8, 21] utilizes vision transformer to encode the image and attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.

More recently, a novel vision and language pretrained model called CLIP [30], which was trained on 400 million real-world text-image pairs by using contrastive learning algorithm. The semantic similarity representation learning by CLIP could be used for zero-shot applications and instructions, especially for the language-guide image editing or construction [13, 29]. Different from the aforementioned work StyleCLIP [29] combines the high-quality images generated by StyleGAN [13], with the rich multi-domain semantics learned by CLIP, our task is to edit videos frame by frame with the help of latent codes for images by the GAN inversion and the extra temporal photometric loss.

3. Approach

3.1. Overview

Direct editing. Given an input video $V_{input} = \{I_1, \cdots, I_T\}$ of T frames, our goal is to semantically edit

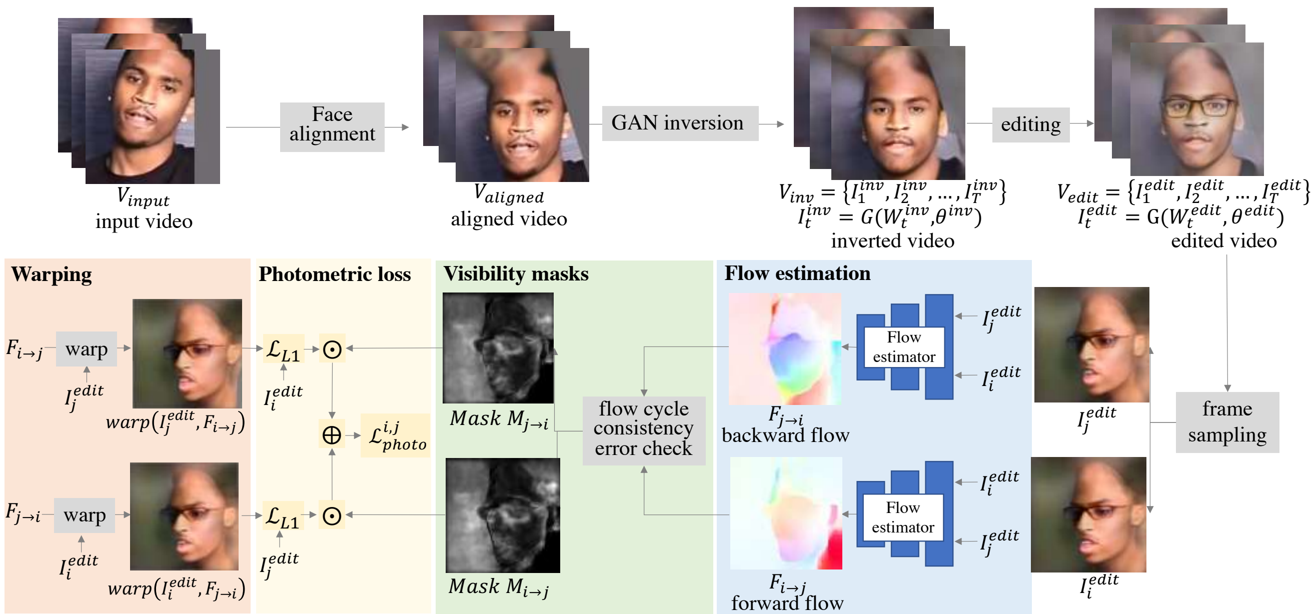

Figure 1. **Video editing with flow-based temporal consistency.** Given an input video of T frames V_{input}, we first spatially align the video frames using off-the-shelf face landmark detector and use existing GAN inversion techniques [3, 31] to obtain the inverted frames $\{I_1^{inv}, I_2^{inv}, \cdots, I_T^{inv}\}$ and their corresponding latent code vectors in the $\mathcal{W}$-space of StyleGAN $\{W_1^{inv}, W_2^{inv}, \cdots, W_T^{inv}\}$. We then independently perform semantic editing on these inverted frames to obtain $\{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$. To achieve temporal coherence, we hierarchically sample two frames I_i^{edit} and I_j^{edit} from V_{edit} at a time. For each sampled frame pair I_i^{edit} and I_j^{edit}, we compute the forward and backward flows $F_{i\to j}$ and $F_{j\to i}$ using RAFT [34]. We then use these two flow fields to compute the visibility masks by performing a forward-backward and backward-forward flow consistency error check. We also use the flows to warp the directly edited frames (*i.e.* $warp(I_i^{edit}, F_{j\to i})$ and $warp(I_j^{edit}, F_{i\to j})$). We finally compute the photometric loss (Eqn. 4) for all possible pairs i and j.

all the video frames while preserving the temporal coherence of the final edited video. To edit the input video V_{input}, we first align its frames and use existing GAN inversion techniques (*e.g.*, [3, 31]) to invert the frames back to the latent codes such that the inverted frames $I_t^{inv} = G(W_t^{inv}; \theta^{inv})$ is similar to the input frame: $I_t^{inv} \approx I_t$. With the inverted frames, we can then edit the inverted video $V_{inv} = \{I_1^{inv}, I_2^{inv}, \cdots, I_T^{inv}\}$ by *independently* editing its frames I_t^{inv}. We denote this frame-by-frame editing approach as "direct editing".

Direct editing on a video. When applying StyleCLIP mapper [29] to a video independently for each frame, we obtain an edited video $V_{edit} = \{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$. For each directly edited frame I_t^{edit}, there is a corresponding latent code W_t^{edit} such that $I_t^{edit} = G(W_t^{edit}; \theta^{edit})$. Due to the per-frame, independent process, the edited video V_{edit} often suffers from temporal inconsistency.

Overview of our approach. We propose a *flow-based* approach to force the edited video to be temporally consistent to address this issue. Our goal is to ensure that the edited video remains temporally consistent while preserving the details from the direct editing. This is inherently a trade-off. On the one hand, preserving all the details from semantic editing leads to temporal flickering artifacts. On the other hand, minimizing the flow-based photometric loss leads to blurry frames (because any blurry frames induce low photometric losses). To address this, we present a test-time optimization approach to balance these two conflicting objectives. Figure 1 shows an overview of our approach. Our core idea is to update the latent code so that the generated frames are temporally consistent (minimizing photometric errors across frames). To do so, we use a flow network to compute the dense bi-directional dense flow map and use them to warp the frames. As all the computations, including the GAN generator, flow estimation network, spatial warping, and photometric losses, are *differentiable*, we can backpropagate the errors all the way back to optimize the latent code. In the following, we describe the detailed procedure and the losses for our approach.

3.2. Flow-based temporal consistency

We propose a flow-based approach to enforce the directly edited video V_{edit} to be temporally consistent.

Hierarchical pair-based optimization. As we cannot fit an entire video into the GPU memory, we choose to optimize the latent code from *pair of frames* at a time. At each iteration, we sample two latent code, W_i^{edit} and W_j^{edit}, corresponding to two frames from V_{edit}, I_i^{edit} and I_j^{edit}. An intuitive way is to use *all* possible pair combinations in a video, but this brings high computation costs. Instead, we use a hierarchical sampling strategy, similar to [16, 28], to reduce the computational costs. The sampled pairs include short-term and long-term pairs:

$$P = \{(I_i^{edit}, I_j^{edit}) \Big| |i-j| = k, k = 1, 2, 4, 8, ...\} . \quad (1)$$

Flow-based consistency. Two sampled frames I_i^{edit} and I_j^{edit} are used to compute the forward and backward flows $F_{i\to j}$ and $F_{j\to i}$ using RAFT [34]. We then use these two flows to warp the directly edited frames (*i.e.* $warp(I_i^{edit}, F_{j\to i})$ and $warp(I_j^{edit}, F_{i\to j})$), and to compute the visibility masks $M_{j\to i}$ and $M_{i\to j}$. Visibility mask $M \in [0, 1]$ is used to penalize the occluded parts. It shows lower weights for occluded pixels and higher weights for the non-occluded pixels. (Shown in Fig 1). To compute the visibility masks, we first compute forward-backward, and backward-forward flow consistency error maps $\epsilon_{j\to i}$ and $\epsilon_{i\to j}$. The error map is computed by $\epsilon_{i\to j}(p) = ||p - F_{j\to i}(p + F_{i\to j}(p))||_2$, where p is a pixel in the flow field. The error map reveals the occlusion degree. Then these resultant error maps are mapped to $[0, 1]$ using an exponential function such that $M_{j\to i} = \exp(-\epsilon_{j\to i})$ and $M_{i\to j} = \exp(-\epsilon_{i\to j})$. Finally, we compute the loss based on the warped frames $warp(I_i^{edit}, F_{j\to i}), warp(I_j^{edit}, F_{i\to j})$ with the visibility masks.

3.3. Two-stage optimization strategy

We split our optimization into two stages. In the first stage, only the latent codes $\{W_t^{edit}\}$ are updated. While in the second stage, we update generator weights θ^{edit}.

Motivation. Our motivation for using the two-stage optimization is that we observe that changing $\{W_t^{edit}\}$ with a small learning rate reflects a *local appearance change*. On the other hand, updating the generator introduces a global change, *e.g.*, the style. We thus split the optimization into two stages to pay special attention when it comes to different editing types.

For the first stage (latent code update), our goal is to update the latent code to minimize the photometric loss:

$$\underset{\{W_t^{edit}\}}{\text{argmin}}\, \mathcal{L}_{\{W_t^{edit}\}} = \mathcal{L}_{photo} \, , \quad (2)$$

For the second stage (generator update), our goal is to finetune the generator to minimize:

$$\underset{\theta^{edit}}{\text{argmin}}\, \mathcal{L}_G = \lambda_r \mathcal{L}_r + \mathcal{L}_{photo} \, , \quad (3)$$

where $\mathcal{L}_{photo}$ again is the bi-directional photometric loss, $\mathcal{L}_r$ is the regularization loss, and λ_r is the strength of regularization. Below we present the detailed loss formulation.

3.4. Loss functions

Bi-directional photometric loss. We minimize the bi-directional photometric loss for both stages to achieve a temporally consistent video. This loss measures the difference between two frames. We use it with the visibility masks to calculate the deviation in the non-occluded parts.

$$\begin{aligned}\mathcal{L}_{photo} = &\sum_{I_i^{edit}, I_j^{edit} \in P} M_{i\to j} ||I_j^{edit} - warp(I_i^{edit}, F_{j\to i})||_1 \\ &+ M_{j\to i}||I_i^{edit} - warp(I_j^{edit}, F_{i\to j})||_1 \, .\end{aligned} \quad (4)$$

Intuitively, bi-directional photometric loss ensures colors along the valid (forward-backward or backward-forward consistent) vectors across frames are as similar as possible. The use of a visibility map helps alleviate the negative impact from occlusion/disocclusion.

Local generator regularization. In the second stage (generator update), we do not wish to *ruin* the pretrained latent space. Therefore, we introduce a regularization loss to help prevent generator G from losing its latent space editability. Similar to [31], we use a *local regularization* to preserve the editing ability of our generator. More specifically, we first obtain a latent code W_r by linearly interpolating between the current latent code W_t^{edit} and a randomly sampled code W_z with an interpolation parameter α_{interp}. This gives us a new latent code in a local region around W^{edit}.

To ensure that we do not lose the editing capability of the original generator, we add penalty on the distance between the generated image from new generator and old one as:

$$\mathcal{L}_r = \mathcal{L}_{LPIPS}(x_r, x'_r) + \lambda^r_{\ell_2} \mathcal{L}_{\ell_2}(x_r, x'_r) \, , \quad (5)$$

where $x_r = G(W_r; \theta^{edit}), x'_r = G(W_r; \theta^{out})$, $\lambda^r_{\ell_2}$ is the weight for ℓ_2 loss. This regularization can alleviate the side effect from updating G, in a local area. Since for a video, the latent codes for the same identity tend to gather locally.

Overall loss function. Our final loss is

$$\theta^{out}, \{W_t^{out}\} = \underset{\theta^{edit}, \{W_t^{edit}\}}{\text{argmin}}\, \lambda_r \mathcal{L}_r + \mathcal{L}_{photo} \, , \quad (6)$$

where θ^{out} is the output parameters of our output generator, and $\{W_t^{out}\}$ are the latent codes of the output video. We use $\lambda^r_{\ell_2} = 1.0, \lambda_r = 200.0, \alpha_{interp} = 30.0$ in all of our experiments.

4. Experiments

4.1. Setup

We test our approach on two sub-sampled datasets, RAVDESS [27] and Voxceleb2 [6]. We subsapmled 10 videos from RAVDESS and 40 videos from Voxceleb2, re-

Figure 2. **Visual results on RAVDESS [27] and Voxceleb2 [6].** Our results maintain consistent changes with time preserving the temporal coherence.

Table 1. **Quantitative Results**

Datasets	RAVDESS	Voxceleb2	RAVDESS	Voxceleb2
Metrics	$E_{warp}\downarrow$		LPIPS↓	
Direct editing	0.0198	0.0234	0.0000	0.0000
angry	0.0027	0.0058	0.2487	0.1248
beard	0.0033	0.0049	0.2914	0.1523
eyeglasses	0.0042	0.0044	0.1241	0.1742
Depp	0.0023	0.0052	0.2521	0.1602
surprised	0.0042	0.0039	0.2018	0.1461
Average perfomance	0.0033	0.0048	0.2236	0.1515

spectively.

For the editing directions, we use StyleCLIP [29] and train five mappers, "eyeglasses", "angry", "surprised", "Depp" and "beard".

We first use Restyle encoder [3] as the GAN inversion method frame by frame, then we apply the directions learned by StyleCLIP frame by frame. We denote this frame-by-frame editing as "direct editing" results. Finally, we apply our approach to the direct editing results.

We evaluate the results with two metrics: 1) temporal consistency and 2) perceptual similarity with the semantically edited frames. To evaluate temporal consistency, we measure the *Warping Error* E_{warp}:

$$E_{warp}(I_t, I_{t+1}) = \frac{1}{\sum_{i=1}^{N} M_t(p_i)} \cdot \sum_{i=1}^{N} M_t(p_i)||I_t(p_i) - \hat{I}_{t+1}(p_i)||_2^2\,, \tag{7}$$

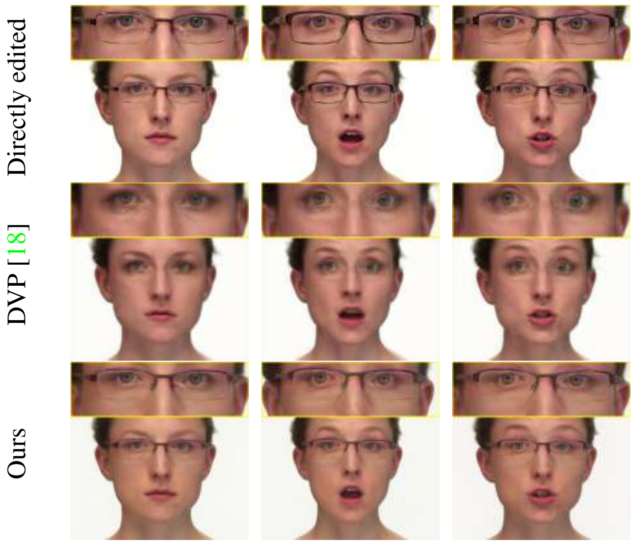

Figure 3. **Visual comparison with DVP [18].** In contrast to our approach, DVP removes the eyeglasses in order to maintain temporal consistency.

where $\hat{I}_{t+1} = warp(I_{t+1}, F_{t\to t+1})$, N is the number of pixels, p_i is the i-th pixel, and M_t is a binary non-occlusion mask, which shows non-occluded pixels. We compute the forward-backward consistency error and use the threshold in [17, 26] to get M_t.

We also measure the perceptual similarity score [37] between the directly edited video $V^{edit} = \{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$ and our output video $V^{out} = \{I_1^{out}, I_2^{out}, \cdots, I_T^{out}\}$ by measuring the averaged perceptual similarity between the corresponding frames:

$$LPIPS = \frac{1}{T}\sum_{t=1}^{T} LPIPS(I_t^{edit}, I_t^{out}), \qquad (8)$$

4.2. Results on RAVDESS

Setup. We perform our two-stage optimization approach on the directly edited video using Adam optimizer [15]. For the second stage, we set the learning rate of G to $\alpha_G = 5 \times 10^{-6}$, and finetune G for ten epochs. We set the regularization weight λ_r to 200.

Analysis. Table 1 shows that our approach improves the temporal consistency over the directly edited video baseline by a large margin. The main challenging source of inconsistency comes from the details of the newly added attributes, *e.g.*, glasses, and some background flickering. We show sample visual results in Figure 2, where the introduced changes are consistent among the different frames.

4.3. Results on VoxCeleb2

Setup. We perform our two-stage optimization approach on the directly edited video using Adam optimizer [15]. For the first stage of training, we set the learning rate of the latent codes to $\alpha_w = 5 \times 10^{-5}$, and update the latent codes for 20 epochs. For the second stage, we set the learning rate of G to $\alpha_G = 5 \times 10^{-6}$, and finetune G for ten epochs. We set the regularization weight λ_r to 200.

Analysis. We observe that our approach does not perform well on Voxceleb2 in Figure 2. This could come from two reasons: a) low resolution of the videos, and b) complicated background. Low-resolution brings worse results for the GAN inversion, while more complicated background brings more redundant pixels for the flow estimation and the optimization.

4.4. Qualitative comparison with DVP

We also compare our results with a blind video consistency approach, DVP [18] visually. The result is shown in Figure 3. We observe that DVP usually sacrifice the details to achieve a temporal consistency.

5. Conclusion

We have presented a novel method for video-based semantic editing by leveraging image-based GAN inversion and editing frame by frame. To deal with the challenges of the temporal constraint, we apply hierarchical pair-based optimization by minimizing the bi-directional photometric loss and the local generator regularization. From our experiments, we show that our method can achieve temporal consistency and preserve its similarity to the direct editing results.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 1

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 1

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2, 4

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1

[5] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10638–10647, 2020. 1

[6] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018. 3, 4

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional

transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[9] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021. 1

[10] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055, 2017. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[12] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 1

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 3

[17] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *European Conference on Computer Vision*, 2018. 5

[18] Chenyang Lei, Yazhou Xing, and Qifeng Chen. Blind video temporal consistency via deep video prior. In *Advances in Neural Information Processing Systems*, 2020. 5

[19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 1

[20] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023. 1

[21] Fuxiao Liu, Hao Tan, and Chris Tensmeyer. Documentclip: Linking figures and main body text in reflowed documents. *arXiv preprint arXiv:2306.06306*, 2023. 1

[22] Fuxiao Liu, Yinghan Wang, Tianlu Wang, and Vicente Ordonez. Visual news: Benchmark and challenges in news image captioning. *arXiv preprint arXiv:2010.03743*, 2020. 1

[23] Fuxiao Liu, Yinghan Wang, Tianlu Wang, and Vicente Ordonez. Visualnews: Benchmark and challenges in entity-aware image captioning. *arXiv preprint arXiv:2010.03743*, 2020. 1

[24] Fuxiao Liu, Yinghan Wang, Tianlu Wang, and Vicente Ordonez. Visual news: Benchmark and challenges in news image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6761–6771, 2021. 1

[25] Fuxiao Liu, Yaser Yacoob, and Abhinav Shrivastava. Covid-vts: Fact extraction and verification on short video platforms. *arXiv preprint arXiv:2302.07919*, 2023. 1

[26] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020. 5

[27] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018. 3, 4

[28] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (TOG)*, 39(4):71–1, 2020. 3

[29] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021. 1, 2, 4

[30] 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. *arXiv preprint arXiv:2103.00020*, 2021. 1

[31] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021. 2, 3

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[33] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 1

[34] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. *arXiv preprint arXiv:2003.12039*, 2020. 2, 3

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1

[36] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021. 1

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5