The DEVIL is in the Details:
A Diagnostic Evaluation Benchmark for Video Inpainting

Ryan Szeto and Jason J. Corso
University of Michigan
{szetor, jjcorso}@umich.edu

Abstract

Quantitative evaluation has increased dramatically among recent video inpainting work, but the video and mask content used to gauge performance has received relatively little attention. Although attributes such as camera and background scene motion inherently change the difficulty of the task and affect methods differently, existing evaluation schemes fail to control for them, thereby providing minimal insight into inpainting failure modes. To address this gap, we propose the Diagnostic Evaluation of Video Inpainting on Landscapes (DEVIL) benchmark, which consists of two contributions: (i) a novel dataset of videos and masks labeled according to several key inpainting failure modes, and (ii) an evaluation scheme that samples slices of the dataset characterized by a fixed content attribute, and scores performance on each slice according to reconstruction, realism, and temporal consistency quality. By revealing systematic changes in performance induced by particular characteristics of the input content, our challenging benchmark enables more insightful analysis into video inpainting methods and serves as an invaluable diagnostic tool for the field. Our code and data are available at github.com/MichiganCOG/devil.

1. Introduction

Video inpainting, *i.e.*, the task of filling in missing pixels in a video with plausible values, pushes the boundaries of modern video editing techniques and enables remarkable applications for film and social media such as watermark and foreground object removal [1, 29]. Compared to image inpainting, video inpainting is more challenging due to the additional temporal dimension, which not only increases the complexity of the solution space, but also places additional constraints on what constitutes a high-quality prediction—in particular, predictions must be coherent in terms of both spatial structure *and* motion. Despite the difficulty of the task, modern results have become quite compelling thanks to the increasing amount of attention that the problem has received as of late [4, 9, 16, 20, 25, 28, 30, 31].

Quantitative evaluation has increased dramatically among recent video inpainting work; however, existing evaluation schemes underemphasize the importance of the contents of the videos and masks used to gauge performance. Typically, video inpainting is evaluated as a reconstruction problem: performance is quantified by masking out arbitrary regions from the video (*i.e.*, "corrupting" it) and scoring the model's ability to recover the masked-out values [4, 18, 28]. However, the difficulty of reconstruction depends on the mask's shape and motion, as well as the content present in the "uncorrupted" video. For example, given a static mask, it is harder to inpaint a video captured by a fixed camera than one captured by a moving camera. In the former case, the region beneath the mask is never visible, so the model effectively needs to "hallucinate" its appearance; in the latter case, the model could transfer appearance information from other frames, a strategy that lies at the heart of many video inpainting approaches [14, 18, 20, 26].

The difficulty of video inpainting is inherently tied to the content of the videos and masks being inpainted; with this principle in mind, we push for more emphasis on content-informed diagnostic evaluation, which can help identify the strengths and weaknesses of modern inpainting methods and improve ablative analysis. To date, the videos used for evaluation have been underappreciated in this regard, having been sourced from datasets for other tasks (*e.g.*, facial analysis [22, 23] and object localization [21, 27]) rather than selected to represent important *inpainting* scenarios. In particular, they contain biases that are essential for the original task, but hinder fine-grained analysis for video inpainting. For example, object localization videos consistently include prominent, moving foreground objects; as a result, standard inpainting evaluation schemes inevitably underrepresent performance on foreground-free videos. Furthermore, other types of motion, *e.g.*, camera and background scene motion, noticeably impact video inpainting performance, but are not controlled for in standard datasets.

In this work, we propose the Diagnostic Evaluation of Video Inpainting on Landscapes (DEVIL) benchmark. It

	Low (*-l)	High (*-h)	
BG scene motion (bgsm-*)			
Camera motion (cm-*)			
FG displacement (fgd-*)			
FG pose motion (fgpm-*)			
FG size (fgs-*)			

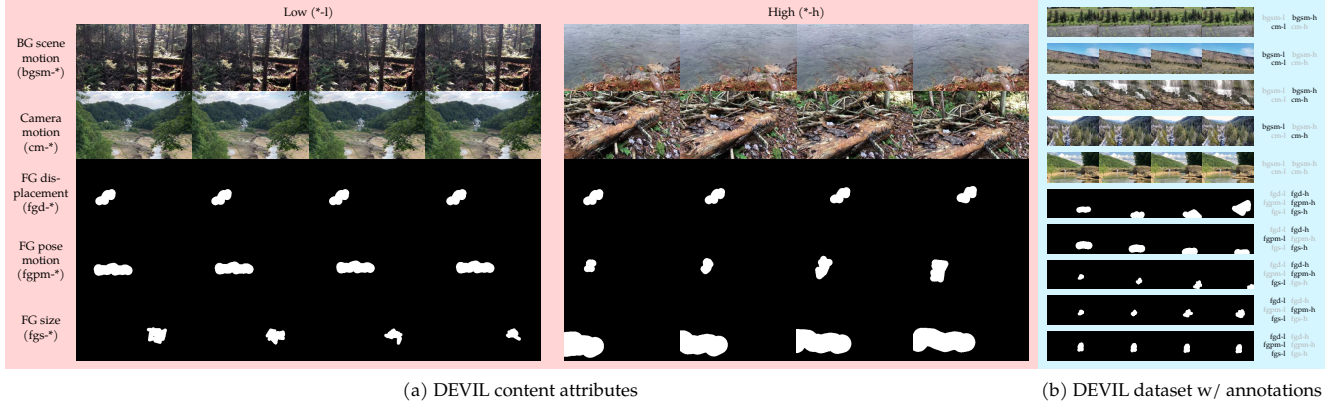(a) DEVIL content attributes (b) DEVIL dataset w/ annotations

Figure 1: A visual overview of our DEVIL dataset. (a) The content attributes that characterize our dataset and are used to create dataset slices for evaluation (*i.e.*, sets of video-mask pairs with a fixed attribute). We label low/high background scene motion or camera motion for videos exhibiting these attribute settings beyond a certain threshold (Section 4.2). For occlusion masks, we construct sampling parameters that capture the desired attribute settings and use them to render masks (Section 4.3). (b) Videos, masks, and annotations from our dataset. A given video or mask may have multiple attribute labels or none; labels for the same attribute are mutually exclusive (*e.g.*, a mask cannot have both low and high FG displacement).

is composed of two parts—the DEVIL dataset and the DEVIL evaluation scheme—which combine to enable a finer-grained analysis than has been possible in prior work. Such granularity is achieved through *content attributes*, *i.e.*, properties of source videos or masks that characterize key failure modes by affecting how easily video inpainting models can borrow appearance information from nearby frames. Specifically, the DEVIL dataset contains source videos labeled with low/high camera and background scene motion attributes, and occlusion masks labeled with low/high foreground displacement, pose motion, and size attributes (Figure 1). Meanwhile, the DEVIL evaluation scheme constructs several *slices* of the DEVIL dataset—sets of video-mask pairs in which exactly one content attribute is kept fixed—and summarizes inpainting quality through metrics that capture reconstruction performance, realism, and temporal consistency (Section 5.2). By controlling for content attributes and summarizing inpainting quality per attribute across several metrics, our DEVIL benchmark provides valuable insight into the failure modes of a given inpainting model and how mistakes manifest in the output.

We use our novel benchmark to analyze the strengths and weaknesses of seven state-of-the-art video inpainting methods. By quantifying their inpainting quality on ten DEVIL dataset slices under five evaluation metrics, we provide the most comprehensive and fine-grained evaluation of modern video inpainting methods to our knowledge. Our head-to-head, multi-faceted comparisons allow us to draw several important conclusions. For example, we show that video inpainting methods in which time and optical flow are carefully modeled consistently achieve the best performance across several types of input data. We also show that the relative rankings between methods are highly sensitive

to metrics as well as source video and mask content, highlighting the need for comprehensive evaluation. Finally, we show that controlling for source video and mask attributes reveals insightful failure modes which can be traced back to the design of the inpainting method in question.

Our comprehensive diagnostic benchmark enables insightful analysis and serves as an invaluable tool for video inpainting research. To summarize, we provide the following contributions:

- We present the first diagnostic dataset specifically designed for video inpainting to our knowledge, which includes annotations for content-based attributes that represent numerous inpainting failure modes;
- We introduce a novel and comprehensive evaluation scheme that spans ten dataset slices and five evaluation metrics of video inpainting quality;
- We analyze seven state-of-the-art algorithms on our benchmark, providing the most comprehensive quantitative evaluation of video inpainting methods to date; and
- We identify systematic errors among video inpainting methods and highlight directions for future work.

Our benchmark is available at https://github.com/MichiganCOG/devil.

2. Related Work

2.1. Methods

Most video inpainting algorithms borrow visual appearance information from known parts of the video to fill in unknown parts. For example, *alignment-based methods* compute local or global alignments between neighboring frames, and then propagate pixels across aligned locations

(alignments are found either through classical feature correspondences [7, 10] or deep neural networks [18, 20, 30]). *Patch-borrowing methods* view videos as spatiotemporal tensors and iteratively paste cuboids or voxels from the known region into the unknown region to maximize global coherence [11, 19, 26]. *Flow-guided methods* propagate visual information along the optical flows estimated between consecutive frames, where the flow for unknown regions is computed iteratively or hierarchically based on known regions to improve performance [9, 14, 16, 28].

Although explicit appearance-borrowing is not leveraged for some methods, *e.g.*, *autoencoder-based methods* [4, 5, 25, 31], we use its prevalence to guide our experimental design. Specifically, we control for content-based attributes that affect the difficulty of borrowing relevant appearance information from other frames, which is possible through our varied data and comprehensive annotations.

2.2. Datasets

The source video datasets used in early video inpainting work were small and oriented around qualitative analysis [11, 19, 26]; as a result, it was difficult to compare the performance of early methods across a wide variety of scenarios. Recent video inpainting methods have instead used large-scale datasets—ranging in structure from aligned face videos [22] and driving videos [6] to unconstrained videos from the Internet [21, 27]—to enable more comprehensive analysis. The foreground object segmentation datasets DAVIS [21] and YouTube-VOS [27] have been particularly popular since their annotations can be used to remove the object from the video. Unlike most prior work, we collect novel videos specifically for the inpainting task, emphasizing attributes that affect how well video inpainting models can transfer appearance information across frames. We also use videos of background scenes instead of videos with foreground objects, which enables us to isolate failure modes that are unrelated to foreground motion.

In terms of occlusion masks, static rectangles [25, 28] or foreground masks from other videos [16, 18, 20, 31] are commonly used. Procedural mask generation has also been explored; for instance, Chang *et al*. [4, 5] render several strokes on a canvas, where each stroke has several control points that randomly move with a certain probability. We extend their work by adding physical constraints for finer control over mask size and motion. Furthermore, we choose our mask sampling parameters based on attributes that directly influence how easily video inpainting models can transfer appearance information across frames.

3. Overview of the DEVIL Benchmark

Before introducing the DEVIL benchmark for video inpainting, we first define the task itself. Let $V \in \{0, \ldots, 255\}^{H \times W \times 3 \times T}$ be an input RGB video with T frames and a resolution of $W \times H$. V contains a placeholder value for missing voxels (*e.g.*, 0) whose locations are indicated by an input occlusion mask $M \in \{0, 1\}^{H \times W \times T}$. Video inpainting aims to produce an inpainted version of V, denoted V^*, with the following characteristics:

- **Reconstruction performance:** V^* is a faithful reconstruction of V^{gt} in the scenario where V is a "corrupted" video derived from some uncorrupted ground truth source video V^{gt}.
- **Realism:** V^* is indistinguishable from a real video.
- **Temporal consistency:** V^* exhibits minimal temporal flickering artifacts.

These criteria are defined more rigorously in Section 5.2.

Our DEVIL benchmark is a collection of tools designed to provide a detailed understanding of video inpainting methods and their behavior across a variety of input data. There are two major components of our benchmark: (i) the DEVIL dataset, which contains source videos and occlusion masks that have been specially curated, rendered, and annotated to identify specific failure modes in video inpainting; and (ii) the DEVIL evaluation scheme, which reports a set of quality-based metrics on several "slices" of the DEVIL dataset, each of which represents a particular failure mode.

The DEVIL dataset captures content complexity along five video-level *content attributes*, *i.e.*, properties that affect the difficulty of inpainting a given video-mask pair by influencing the relevance and availability of appearance information from nearby frames. Specifically, it contains source videos with low and high camera and background (BG) scene motion, and occlusion masks with low and high foreground (FG) displacement, pose motion, and size (Figure 1a).[1] Furthermore, videos and masks are annotated with these attributes and their settings (low or high) to enable targeted evaluation that controls for their presence. Section 4 rigorously defines these attributes and describes our process for collecting videos, masks, and attribute annotations.

Meanwhile, our DEVIL evaluation scheme gauges inpainting quality under multiple *slices* of our dataset, *i.e.*, sets of video-mask pairs characterized by a certain attribute setting. Within each slice, exactly one dataset attribute is kept fixed while the others change freely. By measuring inpainting quality across several slices and metrics, our benchmark provides valuable information on when and how models fail. In Section 5, we describe our DEVIL dataset slices and evaluation metrics in further detail.

4. The DEVIL Dataset

4.1. Collecting Source Videos for the DEVIL

In the context of quantitative evaluation, video inpainting is generally posed as a reconstruction problem [4, 18, 28];

---

[1] The term "foreground" (FG) comes from FG object removal applications.

thus, it is useful to evaluate on videos of background scenes without foreground objects, where the complete ground-truth background appearance is known (and foreground behavior can be controlled explicitly via occlusion masks). Data from other video understanding tasks do not satisfy this criterion, since they generally feature foreground objects which ground the original task. This is especially true for the two most popular datasets used in video inpainting work, DAVIS [21] and YouTube-VOS [27], which were originally collected for foreground object segmentation.

For this reason, we collect our own videos of background-only scenes, similar to Zhang *et al.* [31]. In particular, we target scenic landscape videos in which people have filmed natural outdoor locations from both casual and cinematic viewpoints. Because the primary subject of these videos is the background, they are substantially less likely to contain prominent foreground objects, and are thus good targets for curating our source video collection.

To collect scenic landscape videos, we first search Flickr [15] using the query term "scenic", and retain videos from a fixed set of users who have primarily uploaded high-quality, non-post-processed content between 2017-2019. Then, we apply a combination of automated and manual filtering to remove videos with foreground objects or shot transitions. Finally, we split the filtered videos into clips containing between 45-90 frames, constituting a total of 1,250 clips (examples are shown in Figure 1). Additional details are provided in the supplementary materials.

4.2. Annotating DEVIL Source Video Attributes

For our DEVIL source videos, we annotate two types of content attributes: camera motion and BG scene motion (Figure 1a). **Camera motion** encompasses frame-to-frame differences that are induced by changes in the camera's pose relative to the scene (*i.e.*, camera extrinsics); **BG scene motion** refers to frame-to-frame differences that result from changes in the scene itself, such as running bodies of water or trees that sway due to strong winds (*i.e.*, motion among "stuff" classes in the object detection sense [8]).

We select these attributes for two reasons. First, they represent two sources of complex motion with different low-level characteristics that video inpainting models must replicate well to produce convincing predictions. Second, they impact video inpainting models by influencing the similarity and relevance of appearance information across frames. For instance, high camera motion can reveal or obscure parts of the scene, or otherwise change the scene's appearance due to perspective; high BG scene motion continuously changes the frame-wise appearance of textures.

These attributes are difficult to quantify concretely based on RGB video frames alone; however, it is possible to distinguish extreme examples of low and high motion by visual inspection and proxy estimates. Thus, for a given attribute,

we label videos as containing either low or high motion, but only for a small percentage of videos lying at the extreme ends (videos not labeled for the given attribute may still appear in slices that do not control for it). Not only does this reduce label ambiguity, it also magnifies any performance differences caused by changing a given attribute between low and high settings, thereby highlighting failure modes.

To annotate camera motion, we use classical affine alignment techniques and measure the amount of invalid pixels introduced via warping as a proxy for camera motion; we then threshold the result on either side to produce low and high motion labels. As for BG scene motion, we manually assign low and high labels based on the percentage of the field of view that contains large running bodies of water. Further details are provided in the supplementary materials.

4.3. DEVIL Masks and Attributes

For occlusion masks, we consider three attributes that influence the availability of relevant appearance information in nearby frames (Figure 1a):

- **FG displacement:** How much the mask's centroid moves over time with respect to the field of view;
- **FG pose motion:** How much the shape of the mask changes over time with respect to the field of view (independent of the displacement of its centroid); and
- **FG size:** The average number of pixels that are occupied by the mask per frame.

Masks with high FG displacement or pose motion reveal complementary parts of the scene over time, whereas FG size explicitly determines how much appearance information can be relied on as ground truth.

To generate masks with low and high settings for these attributes, we adapt the procedural blob generation strategy by Chang *et al.* [4]. In particular, we adjust their stroke width, velocity, and stochasticity parameters to correspond to low or high FG displacement, pose motion, and size. For more details on the mask generation process, including our extensions to enable finer control over individual blobs, refer to the supplementary materials.

5. The DEVIL Evaluation

5.1. Slices of the DEVIL Dataset

The naïve way to evaluate on the DEVIL dataset would be to randomly sample a test set of paired source videos and occlusion masks without accounting for their attributes; however, this provides little insight into the failure modes that cause prediction errors for a given method. Instead, we control for one attribute at a time to isolate its impact on the prediction. Specifically, for each attribute setting, we construct *slices* of the DEVIL dataset, *i.e.*, pre-determined sets of video-mask pairs where the given attribute is fixed as low or high and the others are uncontrolled. By reporting
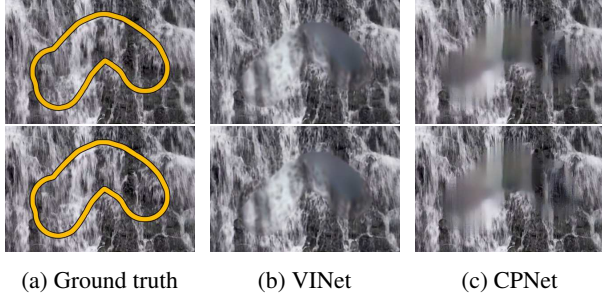
(a) Ground truth (b) VINet (c) CPNet

Figure 2: High temporal consistency may indicate overly blurry predictions if reconstruction or realism performance is low (as shown in the results of VINet and CPNet). The area to be inpainted is outlined in yellow in (a).

performance on each slice separately, our benchmark can highlight failure modes with finer granularity.

We construct the DEVIL dataset slices as follows. Given the desired attribute setting (*e.g.*, low camera motion), we randomly sample either 150 source videos or masks with that setting (recall that an attribute applies exclusively to either the source video or the mask modality). Then, within the other modality, we sample 150 instances from all available DEVIL instances (*e.g.*, for the low camera motion slice, we sample all rendered DEVIL masks). Finally, we pair together the selected source videos and masks.

5.2. Evaluation Metrics

We evaluate on composited inpainting results (*i.e.*, the known region is composited over the inpainting prediction). Inputs are resized to 832×480 resolution; for methods that cannot consume this resolution, we apply mirror padding to both the source video and the mask, run the method, and crop out padded regions from the result for fairness. We quantify performance along three axes of inpainting quality—reconstruction, realism, and temporal consistency—as described in the remainder of this section.

Reconstruction captures the extent to which a video inpainting method predicts the original content in a given reference video (*i.e.*, the version of the video without the occlusion mask). We report two reconstruction metrics: the Learned Perceptual Image Patch Similarity (LPIPS) metric [32] with a pre-trained AlexNet backbone [17], and our own video-based variant called the Perceptual Video Clip Similarity (PVCS) metric with a pre-trained I3D backbone [3]. These metrics measure the distance between corresponding features of a deep neural network. LPIPS is computed between corresponding frames of the reference and inpainted video, whereas PVCS is computed between corresponding 10-frame clips in a sliding window.

Realism indicates how well the inpainting result resembles the appearance and motion observed in a reference set

of real videos, independent of the original video from which the input is derived. Unlike reconstruction, realism enforces a smaller penalty for deviating from the original content as long as the prediction exhibits sensible appearance and motion. For our image-based realism metric, we use the Fréchet Inception Distance [13] (FID), which fits multivariate normal distributions over the feature activations of two sets of images and measures their distance; in our case, the two sets correspond to all predicted frames and all reference frames. We also report the video-based equivalent Video FID [16] (VFID), which corresponds to the sets of all inpainted videos and all reference videos (the features are extracted from a pre-trained I3D backbone [3]).

Temporal consistency measures the proliferation of flickering artifacts, *i.e.*, how much colors at corresponding points of the scene change between consecutive frames. We adapt the patch consistency metric, denoted PCons, from Gupta *et al.* [12]: for each frame, we extract the 50×50 patch at the centroid of the occlusion mask, compute the maximum Peak Signal-to-Noise Ratio (PSNR) between this patch and neighboring patches in the next frame, and average the result across all frames. Note that stronger temporal consistency is not always ideal: low-quality predictions, such as constant-color or blurry inpainting results, can produce high temporal consistency scores (see Figure 2).

6. Experiments

To demonstrate the utility of our DEVIL benchmark, we analyze the performance of seven representative state-of-the-art video inpainting methods, using the publicly-available code, model weights, and default runtime arguments provided by the original authors:

- **Joint optimization of flow and color (JointOpt) [14]:** Alternates between optimizing an optical flow estimate and finding suitable patches along the flow. Among our methods, this is the only non-deep learning one.
- **VINet [16]:** Recurrently predicts the next frame by warping intermediate features spatially via optical flow.
- **Deep Flow Completion Network (DFCNet) [28]:** Predicts the optical flow of the video, then inpaints the missing region by propagating known values along the flow.
- **Copy-Paste Network (CPNet) [18]:** Estimates affine transformations between frames with a task-driven deep neural network, and then copies features across aligned frames via attention.
- **Onion Peel Network (OPN) [20]:** Iteratively inpaints the exterior of the current missing region by attending to relevant locations in the known region.
- **Spatio-Temporal Transformer Network (STTN) [30]:** Decodes the missing region with a Transformer [24] that consumes multi-scale patches from the entire video.
- **Flow-Edge Guided Video Completion (FGVC) [9]:**

Low					FG displacement	High				
LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲	Method	LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲
0.00504	0.1910	7.73	0.0513	36.96	JointOpt	0.00206	0.0995	**1.71**	0.0197	36.07
0.00643	0.3053	21.68	0.1064	47.92	VINet	0.00370	0.2258	8.29	0.0648	44.39
0.00479	**0.1822**	7.29	0.0507	55.81	DFCNet	**0.00190**	0.1028	2.08	0.0241	**56.35**
0.00419	0.2254	12.72	0.0740	42.36	CPNet	0.00236	0.1462	4.85	0.0367	40.33
0.00379	0.1887	**7.29**	**0.0466**	35.14	OPN	0.00253	0.1474	3.46	0.0305	34.60
0.00411	0.2265	8.18	0.0660	39.90	STTN	0.00293	0.1602	3.68	0.0390	38.39
0.00465	0.1914	9.17	0.0495	37.02	FGVC	0.00200	**0.0986**	1.89	**0.0182**	38.15
0.00471	0.2158	10.58	0.0635	42.16	Mean	0.00250	0.1401	3.71	0.0333	41.18

Low					FG pose motion	High				
LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲	Method	LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲
0.00401	0.1643	5.79	0.0419	36.84	JointOpt	0.00296	0.1266	3.85	0.0282	36.56
0.00618	0.2869	17.00	0.0956	46.40	VINet	0.00435	0.2456	10.51	0.0730	46.03
0.00354	**0.1604**	**5.06**	0.0430	54.07	DFCNet	0.00294	**0.1237**	3.38	0.0305	**55.17**
0.00383	0.2060	10.34	0.0655	41.60	CPNet	0.00323	0.1681	6.33	0.0481	41.16
0.00319	0.1794	6.01	0.0433	35.18	OPN	**0.00289**	0.1606	3.68	0.0367	34.54
0.00395	0.2101	6.96	0.0575	39.80	STTN	0.00328	0.1809	4.77	0.0481	39.14
0.00391	0.1634	6.03	**0.0392**	37.58	FGVC	0.00290	0.1261	4.39	**0.0259**	37.75
0.00409	0.1958	8.17	0.0551	41.64	Mean	0.00322	0.1617	5.27	0.0415	41.48

Low					FG size	High				
LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲	Method	LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲
0.00078	**0.0495**	1.99	**0.0102**	36.18	JointOpt	0.00591	0.2324	**7.30**	**0.0530**	37.19
0.00118	0.1094	6.31	0.0288	43.81	VINet	0.00839	0.4312	23.63	0.1488	50.07
0.00069	0.0519	0.93	0.0119	53.80	DFCNet	0.00556	0.2240	7.71	0.0627	**57.80**
0.00073	0.0680	2.35	0.0155	39.17	CPNet	0.00679	0.3048	13.21	0.0969	43.91
0.00084	0.0563	0.95	0.0119	34.30	OPN	0.00571	0.2874	8.71	0.0708	35.09
0.00113	0.0786	2.02	0.0194	38.24	STTN	0.00645	0.3149	10.89	0.0911	41.12
0.00087	0.0528	1.66	0.0119	37.89	FGVC	**0.00543**	0.2322	9.76	0.0564	38.26
0.00089	0.0666	2.32	0.0157	40.48	Mean	0.00632	0.2896	11.60	0.0828	43.35

Low					BG scene motion	High				
LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲	Method	LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲
0.00247	**0.1199**	**3.70**	**0.0257**	39.36	JointOpt	0.00512	0.1680	3.89	**0.0333**	34.91
0.00434	0.2549	13.01	0.0791	47.87	VINet	0.00543	0.2793	14.39	0.0796	**47.02**
0.00275	0.1381	5.97	0.0387	54.00	DFCNet	**0.00362**	0.1620	**3.40**	0.0398	40.00
0.00279	0.1691	6.57	0.0499	43.38	CPNet	0.00398	0.2070	5.24	0.0591	41.48
0.00254	0.1587	4.90	0.0397	36.67	OPN	0.00367	0.1819	4.27	0.0349	33.55
0.00299	0.1873	6.60	0.0569	40.82	STTN	0.00447	0.2030	4.15	0.0438	38.78
0.00314	0.1347	7.70	0.0370	39.16	FGVC	0.00376	0.1697	4.57	0.0356	36.50
0.00300	0.1661	6.92	0.0467	43.04	Mean	0.00429	0.1958	5.70	0.0466	38.89

Low					Camera motion	High				
LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲	Method	LPIPS ▼	PVCS ▼	FID ▼	VFID ▼	PCons ▲
0.00269	0.1311	**3.03**	0.0253	39.39	JointOpt	0.00275	0.1319	**2.36**	0.0220	33.84
0.00610	0.2909	17.87	0.0863	48.39	VINet	0.00467	0.2417	10.44	0.0613	44.00
0.00254	**0.1169**	3.58	**0.0224**	55.22	DFCNet	0.00352	0.1353	3.34	0.0293	**45.13**
0.00344	0.1585	5.25	0.0339	44.04	CPNet	0.00496	0.2227	7.21	0.0697	39.56
0.00357	0.1749	3.64	0.0340	34.81	OPN	0.00327	0.1679	4.60	0.0332	32.35
0.00349	0.1521	4.34	0.0280	41.95	STTN	0.00529	0.2410	8.95	0.0783	36.38
0.00284	0.1378	6.28	0.0258	42.65	FGVC	0.00312	**0.1239**	3.05	**0.0206**	34.61
0.00352	0.1660	6.28	0.0365	43.78	Mean	0.00394	0.1806	5.71	0.0449	37.98

Table 1: The performance of each inpainting method on each DEVIL slice and evaluation metric. Bold indicates the best method; ▼ and ▲ indicate that lower or higher is better, respectively.

Figure 3: Mean performance of each method across all DEVIL slices. Error bars show standard error across the ten slices.

Extends DFCNet [28] by leveraging flow between non-adjacent frames and using edge information to solve for piecewise-smooth flow predictions.

6.1. Aggregate Analysis

In Table 1, we report the performance of all evaluated methods on each DEVIL slice; in Figure 3, we compare their performance averaged across all slices. We observe that JointOpt, DFCNet, and FGVC consistently outrank or perform within one standard error of the other methods in terms of the reconstruction metrics LPIPS/PVCS and the realism metrics FID/VFID. They all explicitly solve for the optical flow of the inpainted video during inference, suggesting that *computing task-driven flow is an essential ingredient in producing the highest-quality video inpainting results*. Additionally, JointOpt remains competitive among recent deep learning-based solutions, suggesting that improvements may arise by adapting traditional subroutines, *e.g.*, PatchMatch [2], to deep learning.

The three mid-tier methods—OPN, STTN, and CPNet—borrow intermediate features across distant time steps, but do not use time as an ordered structure. Meanwhile, VINet models time through a recurrent unit, but has the shortest temporal receptive field among the evaluated methods and cannot propagate information from future time steps back through the entire video. These results indicate that *modeling time as a proper ordered structure with long-range dependencies greatly improves inpainting quality*.

In terms of temporal consistency, DFCNet achieves the highest PCons since it propagates pixel values directly along the predicted flow maps of adjacent frames. Interestingly, JointOpt and FGVC achieve lower PCons despite also propagating values along the flow, likely due to their ability to transfer candidate values from non-adjacent frames.
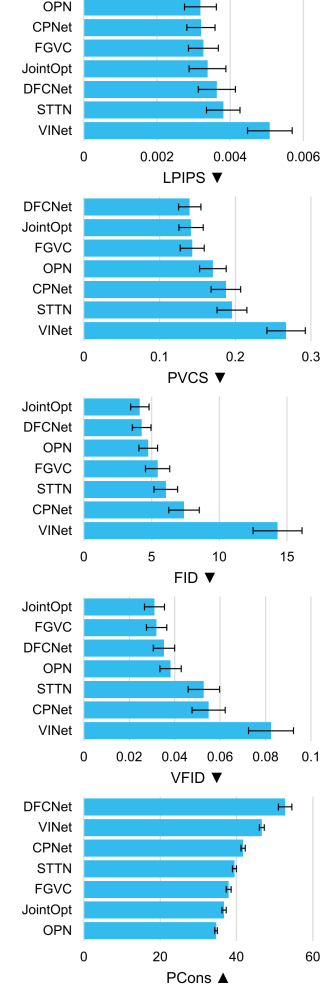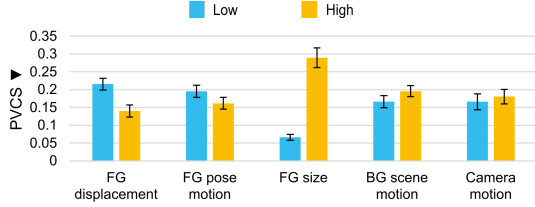
Figure 4: Comparison of DEVIL slice difficulty based on average model performance (lower is better; error bars show standard error across the seven models). The type of content given at test time, especially the mask content, greatly affects the difficulty of the task.

	FG displacement		Camera motion	
	Low	High	Low	High
	OPN	**DFCNet**	**DFCNet**	**JointOpt**
	STTN	FGVC	JointOpt	FGVC
LPIPS	CPNet	JointOpt	FGVC	OPN
	FGVC	CPNet	CPNet	DFCNet
	DFCNet	OPN	STTN	VINet
	JointOpt	STTN	OPN	CPNet
	VINet	VINet	VINet	STTN
	Low	High	Low	High
	OPN	**FGVC**	**DFCNet**	**FGVC**
	FGVC	JointOpt	JointOpt	JointOpt
VFID	DFCNet	DFCNet	FGVC	DFCNet
	JointOpt	OPN	STTN	OPN
	STTN	CPNet	CPNet	VINet
	CPNet	STTN	OPN	CPNet
	VINet	VINet	VINet	STTN

Table 2: Methods sorted by performance from best to worst based on three variables: the metric (LPIPS/VFID), the attribute (FG displacement/camera motion), and the setting of said attribute (low/high). The strongest method (highlighted in bold) depends on all three variables, showing that no one method dominates our challenging benchmark.

VINet and CPNet achieve high temporal consistency at the cost of low-quality predictions lacking appropriate texture or motion (Figure 2); the behaviors of these two models indicate that temporal consistency is most meaningful when reconstruction and realism performance is also high.

DEVIL Attribute Difficulty We now analyze how DEVIL attributes impact overall inpainting difficulty to highlight their utility in video inpainting evaluation. In Figure 4, we compare the difficulty of each DEVIL attribute setting by computing the average PVCS over all methods on the corresponding test slice. The mask attributes substantially impact the difficulty of video inpainting; in particular, higher FG displacement and pose motion, as well as smaller FG size, lead to better performance. These trends make sense—more relevant appearance information is available in other frames when the occlusion mask is smaller and moves more over time. In contrast, the overall impact of camera and BG scene motion is small due to the differing sensitivities of each method to these attributes (Section 6.2). We observe similar trends under the other reconstruction and realism metrics (refer to the supplementary materials for details).
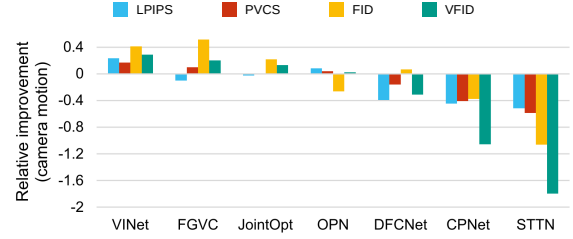

Figure 5: Relative improvement of each method under reconstruction and realism metrics when camera motion changes from low to high.
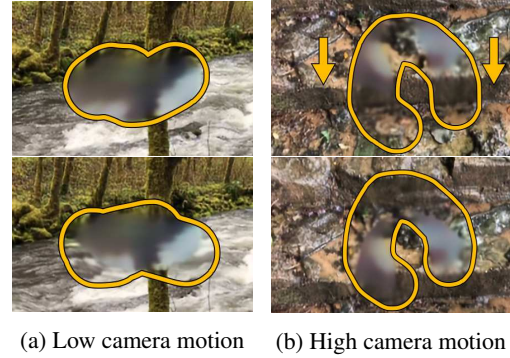

(a) Low camera motion   (b) High camera motion

Figure 6: VINet prediction examples. VINet improves with high camera motion since content "moves into" the missing region (arrows show the direction of camera motion).

6.2. Model Sensitivity to DEVIL Attributes

Next, we compare the performance of each method on individual DEVIL slices to demonstrate the impact of video and mask content, as well as the metric, on their rankings. Table 2 lists the methods sorted by performance and grouped by three variables: (i) the metric, (ii) the DEVIL attribute, and (iii) the particular setting of the attribute (*i.e.*, low or high). Note that the two attributes shown, FG displacement and camera motion, span the two modalities of the DEVIL dataset (masks and videos, respectively); also note that these three variables span the complexity of the DEVIL evaluation scheme in terms of metrics and dataset slices. Even among the eight different combinations of variables shown out of the possible 50, four methods achieve the best performance for at least one combination; this demonstrates that our DEVIL benchmark is a substantial challenge for video inpainting methods.

Finally, we analyze the sensitivity of the methods to each DEVIL attribute by changing the label and plotting the relative difference in performance. Specifically, for a given attribute, we compute the relative improvement as $(\mathrm{score}_{hi} - \mathrm{score}_{lo})/\mathrm{score}_{lo}$ (or the negation of this value if a lower score is better), where score_{hi} and score_{lo} correspond to model performance on the high and low slices of the attribute, respectively. As evidenced by the following obser-
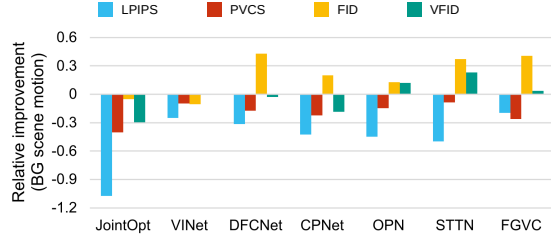
Figure 7: Relative improvement of each method under reconstruction and realism metrics when BG scene motion changes from low to high.
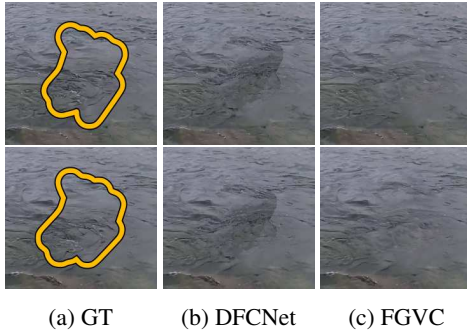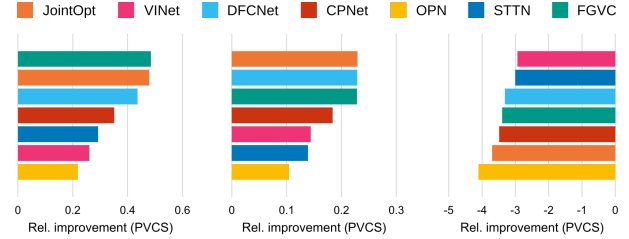

(a) GT (b) DFCNet (c) FGVC

Figure 8: Example inpainting predictions from the high BG scene motion slice. For DFCNet and FGVC, the predictions diverge from the original content, but still exhibit semantically sensible appearance.

vations, our DEVIL attributes enable a fine-grained comparison of failure modes among different inpainting models.

In Figure 5, we see divergent behavior among the methods when camera motion is increased: for example, STTN and CPNet experience dramatic performance drops under all reconstruction and realism metrics, whereas VINet experiences substantial gains. From a design standpoint, the behaviors of STTN and CPNet make sense because they rely on aligning patches or entire frames from other time steps to borrow their features, which can fail catastrophically under heavy camera motion. On the opposite end, VINet's behavior reflects its tendency to inpaint realistic textures only after they move into the missing portion of the frame (Figure 6), which is unlikely to occur without camera motion.

From Figure 7, we see that reconstruction performance under LPIPS and PVCS consistently worsens when BG scene motion increases, reflecting the challenge of replicating complex dynamics precisely. Interestingly, frame realism under FID actually improves dramatically for some methods such as DFCNet and FGVC because they inpaint "alternate realities", *i.e.*, appearance that diverges from the original content, but still captures the original broad structure (Figure 8). Among the methods that explicitly infer and propagate across flow (FGVC, JointOpt, and DFCNet),

(a) FG displacement (b) FG pose motion (c) FG size

Figure 9: Relative improvement of each method under PVCS when mask attributes change from low to high.

only JointOpt achieves worse frame realism performance, suggesting that *video inpainting quality can be improved by simply learning flow in a task-driven, end-to-end manner.*

Moving on to mask attributes, we show in Figure 9a-b that all methods perform better with increased FG displacement and pose motion, indicating that they all leverage the increased availability of background information. The flow propagation methods generally benefit the most, likely due to the explicit transmission of more ground-truth appearance information along predicted flows. We observe similar trends under the other reconstruction and realism metrics (details are available in the supplementary materials).

In Figure 9c, we observe worse performance when the mask FG size grows, reflecting the inherently greater difficulty of inpainting more values. Although this trend is universal and intuitive, the quantitative difference between methods still lends insight into their failure modes. For example, OPN is most sensitive to the increased mask size because it must iteratively inpaint more outer layers of the unknown region based on its own predictions, thereby accumulating error across more iterations.

7. Conclusion

We have presented the DEVIL benchmark for video inpainting and used it to analyze seven state-of-the-art methods, thereby providing the largest fine-grained analysis of video inpainting to our knowledge. By controlling for five content attributes of the source videos and masks used at test time, our analyses have provided novel insight into the behaviors, strengths, and weaknesses of these methods.

Acknowledgements

References

[1] Production Notes: Match Moving | Nevada Film Office, Feb. 2018. Library Catalog: nevadafilm.com.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics*, 28(3):24:1–24:11, July 2009.

[3] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. pages 6299–6308, 2017.

[4] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-Form Video Inpainting With 3D Gated Convolution and Temporal PatchGAN. In *IEEE International Conference on Computer Vision*, Oct. 2019.

[5] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Learnable Gated Temporal Shift Module for Deep Video Inpainting. In *British Machine Vision Conference*, Sept. 2019.

[6] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012.

[7] Mounira Ebdelli, Olivier Le Meur, and Christine Guillemot. Video Inpainting with Short-Term Windows: Application to Object Removal and Error Concealment. *IEEE Transactions on Image Processing*, 24(10):3034–3047, Oct. 2015. Conference Name: IEEE Transactions on Image Processing.

[8] David A. Forsyth, Jitendra Malik, Margaret M. Fleck, Hayit Greenspan, Thomas Leung, Serge Belongie, Chad Carson, and Chris Bregler. Finding Pictures of Objects in Large Collections of Images. In Jean Ponce, Andrew Zisserman, and Martial Hebert, editors, *International Workshop on Object Representation in Computer Vision*, Lecture Notes in Computer Science, pages 335–360, Berlin, Heidelberg, 1996. Springer.

[9] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-Edge Guided Video Completion. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 713–729, Cham, 2020. Springer International Publishing.

[10] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt. Background Inpainting for Videos with Dynamic Objects and a Free-Moving Camera. In *European Conference on Computer Vision*, pages 682–695. Springer, 2012.

[11] M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, and C. Theobalt. How Not to Be Seen — Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum*, 31(2pt1):219–228, 2012.

[12] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and Improving Stability in Neural Style Transfer. In *IEEE International Conference on Computer Vision*, Oct. 2017.

[13] 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. In *Neural Information Processing Systems*, NIPS'17, pages 6629–6640, Long Beach, California, USA, Dec. 2017. Curran Associates Inc.

[14] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally Coherent Completion of Dynamic Video. *ACM Transactions on Graphics*, 35(6):196, 2016.

[15] SmugMug Inc. Flickr. https://www.flickr.com/.

[16] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep Video Inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Neural Information Processing Systems*, 2012.

[18] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-Paste Networks for Deep Video Inpainting. In *IEEE International Conference on Computer Vision*, pages 4413–4421, 2019.

[19] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. Video Inpainting of Complex Scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

[20] Seoung Wug Oh, Sungho Lee, Joon-Young Lee, and Seon Joo Kim. Onion-Peel Networks for Deep Video Completion. In *IEEE International Conference on Computer Vision*, pages 4403–4412, 2019.

[21] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

[22] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics: A Large-Scale Video Dataset for Forgery Detection in Human Faces. *arXiv*, 2018.

[23] Jie Shen, Stefanos Zafeiriou, Grigoris G. Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The First Facial Landmark Tracking In-the-Wild Challenge: Benchmark and Results. In *IEEE International Conference on Computer Vision Workshops*, Dec. 2015.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Neural Information Processing Systems*, 2017.

[25] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video Inpainting by Jointly Learning Temporal Structure and Spatial Details. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 5232–5239, 2019.

[26] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-Time Completion of Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007. Publisher: IEEE.

[27] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-Sequence Video Object Segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 603–619, Cham, 2018. Springer International Publishing.

[28] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep Flow-Guided Video Inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2019.

[29] Charles Yeager. Everything You Need to Know About Chroma Key and Green Screen Footage, July 2019. Library Catalog: www.premiumbeat.com Section: Video Production.

[30] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning Joint Spatial-Temporal Transformations for Video Inpainting. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 528–543, Cham, 2020. Springer International Publishing.

[31] Haotian Zhang, Long Mai, Ning Xu, Zhaowen Wang, John Collomosse, and Hailin Jin. An Internal Learning Approach to Video Inpainting. In *IEEE International Conference on Computer Vision*, Oct. 2019.

[32] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.