TimeReplayer: Unlocking the Potential of Event Cameras for Video Interpolation

研究背景：以高 FPS 记录快速运动需要昂贵的高速相机，作为替代方案，在普通相机中插入低 FPS 视频已经引起了极大的关注，如果只有低 FPS 视频可用，需要运动假设，以推断中间帧，这不能模拟复杂运动。事件相机是一种新型相机，可以在任意复杂运动的情况下实现视频插值，但是由于缺乏算法，潜力尚未得到充分发挥，为了充分发挥事件摄像机的潜力，本文提出了一种新的 TimeReplayer 算法，将商品摄像机捕获的视频与事件进行插值。它以无监督循环一致的方式进行训练，取消了高速训练数据的必要性，并带来了视频外推的额外能力。其先进的结果和演示视频补充揭示了基于事件视觉的美好未来。

研究问题：视频插值是推断两帧之间的中间帧，近些年来，电脑视觉社区在适当假设的方向上进行了很多探索。提出了基于线性运动假设的 SuperSloMo 视频插值方法，并且在二次运动假设的基础上对结果进行了改进。随着假设越来越复杂，出现了两个障碍:(1)线性运动可以在给定两帧的情况下插值视频，但是二次运动需要连续四帧来计算二次插值的加速度。随着假设变得越来越复杂，需要更多的帧和更多的计算成本来插值每一帧。(2)视频中的底层运动可能是任意的，这使得对运动类型的预设假设难以验证。当假设和动作类型不匹配时，插值后的视频可能看起来不真实。为了解决传统的基于帧的摄像机固有的缺乏中间信息的问题，我们引入了一种新的神经形态传感器，事

件摄像机[6,14,33,40]，作为复杂运动情况下视频插值的一种很有前途的解决方案。视频插值任务的另一个关键问题是很难在现实世界场景中收集视频插值的地面事实。大多数现有的视频插值模型[2,9,23,38]都是在高帧率数据集上训练的，该数据集是通过对高速摄像机录制的高帧率视频的连续帧进行平均而构建的。然而，这些模型将受到合成数据和真实世界数据之间的领域差距的影响。因此，让模型具备泛化到没有基础真值的数据的学习能力是很重要的。

想法动机：本文将事件摄像机数据引入到视频插值模型设计中，提出了一种无监督学习框架来训练视频插值模型。具体来说，我们引入了事件流来帮助直接估计中间帧之间的光流输入帧而不是计算它们作为输入帧之间计算光流的比例。这样就打破了均匀线性运动的假设，所提出的光流估计模块可以计算任何复杂的非线性运动。然后根据估计的光流对输入帧进行翘曲，并对其进行加权平均，从而预测出中间帧。在事件流和近似逆事件流的帮助下，视频插值模型能够在给定两个输入帧的情况下预测中间帧，并在给定估计的中间帧和另一个输入帧的情况下重建一个输入帧。因此，可以计算输入帧重建与原始帧之间的损失函数，并用于训练视频插值模型。在合成基准数据集上的大量实验和烧烧研究证明了该框架的有效性，特别是在具有复杂运动的视频上。此外，我们还进一步对实际数据进行了实验，验证其泛化能力。

具体方法：

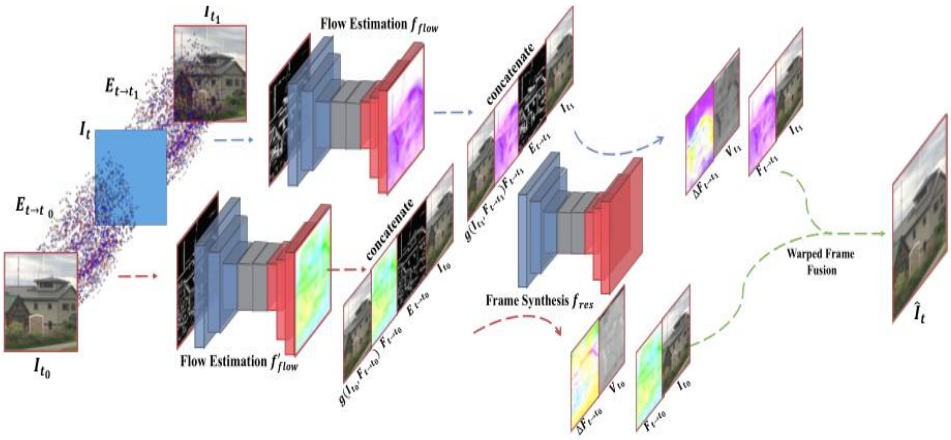


Figure 2. The proposed TimeReplayer model architecture with event stream.

模型基于 SuperSloMo 有三个模型，如下：

流估计：由于事件流本身带有连续的运动信息，因此可以通过将事件表示输入卷积神经网络来直接计算光流。然而，由于近似反向事件流与实际事件流的不同，我们采用网络架构相同但不共享参数的两个流量估计 CNN 模块 f_{flow} 和 f'_{flow} 分别处理正向流事件 $E_{t\to t_1}$ 和反向流事件 $E_{t\to t_0}$ 。所对应的计算光流分别记为 $F_{t\to t_1}$ 和 $F_{t\to t_0}$ 。以往的方法大多假设物体沿直线匀速运动，计算输入帧与目标中间帧之间的光流占两个输入帧之间光流的比例，相比之下，该方法可以更准确地估计非线性运动情况下插值帧与输入帧之间的光流。

流程优化：由于基于事件的光流估计方法往往产生集中在运动物体边缘的稀疏输出。因此，另一个低分辨率 CNN 通过计算残余流ΔF 来细化初始估计的稀疏光流 $F_{t\to t_0}$ 和 $F_{t\to t_1}$ 。最终估计由初始估计和残差之和得到。

$$(\Delta F_{t\to t_0}, V_{t_0}) = f_{res}(g(I_{t_0}, F_{t\to t_0}), I_{t_0}, F_{t\to t_0}, E_{t\to t_0})$$
$$(\Delta F_{t\to t_1}, V_{t_1}) = f_{res}(g(I_{t_1}, F_{t\to t_1}), I_{t_1}, F_{t\to t_1}, E_{t\to t_1}),$$

其中 $g(I_{t_0}, F_{t\to t_0})$ 和 $g(I_{t_1}, F_{t\to t_1})$ 是使用初始估计的光流 $F_{t\to t_0}$ 和$F_{t\to t_1}$ 计算的扭曲输入帧，ΔFt→t0 和ΔFt→t1 表示用于细化光流的残差，Vt0 和 Vt1 表示可见度图，有助于减轻后期混合过程中的伪影和遮挡。将两组特征图分别拼接，并通过共享的低分辨率光流细化 f_{flow_res} 模型进行传递。

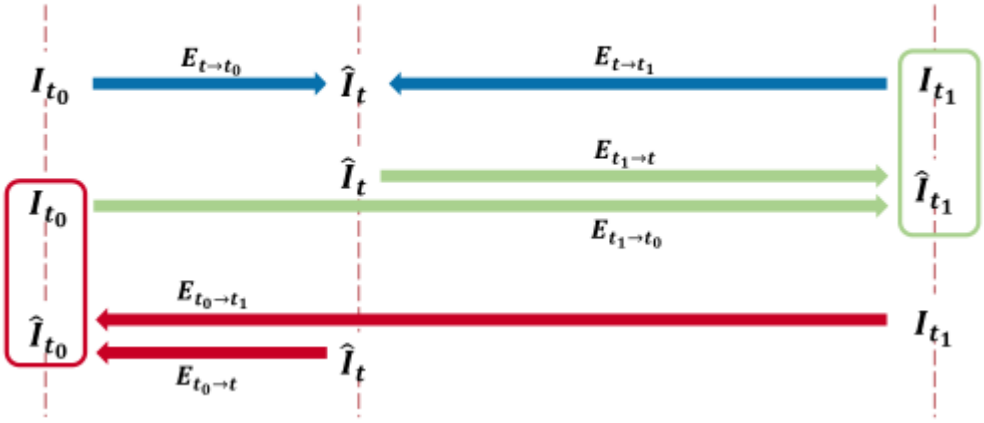
帧的合成：目标帧 $\hat{I}_t$ 可以通过使用精细的光流混合扭曲的输入帧来合成，将混合过程作为两个扭曲帧的加权平均值，以时间间隔和可见度映射的乘积为权重，如式 2 所示。

$$\begin{aligned}\hat{I}_t = &\frac{1}{Z} \odot ((t_1 - t)V_{t_0} \odot g(I_{t_0}, F_{t\to t_0} + \Delta F_{t\to t_0}) + \\ &(t - t_0)V_{t_1} \odot g(I_{t_1}, F_{t\to t_1} + \Delta F_{t\to t_1})).\end{aligned} \tag{2}$$

因此，只要给定 t0 和 t1 两个时间戳的两个输入帧，以及这两个时间戳与目标时间戳 t 之间的事件流，我们就可以使用所提出的整个视频插值模型 f 在该时间戳合成所需的帧。

在该项工作中我们专注于无监督方式训练插值模型，受周期一致性的启发我们提出了一种用于事件数据视频插值的无监督训练框架。在事件数据的帮助下，模型能够在不同时间戳的两个输入帧之间合成一个中间帧；同样，它也可以合成给定预测中间帧的输入帧和另一个输入帧。这样，我们就可以对输入帧的重建进行监督，从而训练视频插值模型。具体来说，对于时间 t0 和 t1 之间的间隔，给定一对输入

帧 I_{t_0} 和 I_{t_1}，事件流 $E_{t_0 \to t_1}$，那么我们可以使用视频插值模型 f 来生成中间帧 $\hat{I}_t$，其中，$\hat{I}_t = f(I_{t_0}, I_{t_1}, E_{t \to t_0}, E_{t \to t_1})$其中，f 表示整个视频插值模型，t∈(t0, t1)表示目标插值时间，$E_{t \to t_0}$ 和 $E_{t \to t_1}$ 分别表示从 t 时刻到 t0 时刻的近似反向流事件和从 t 时刻到 t1 时刻的实际正向流事件。插值过程如下：



实现策略：我们使用 Pytorch 来实现所提出的框架。从相应的连续帧中裁剪 256 × 256 个补丁，批量大小设置为 28。模型在 4 块 NVIDIA Tesla V100 gpu 上使用 Adam 优化器进行训练，其中 β1=0.9，β2=0.999。学习率最初设置为 1e−4，然后每 200 个 epoch 缩小 0.1，直到 500 个 epoch。我们采用峰值信噪比(PSNR)、结构相似度图像度量(SSIM)和插值误差(IE)作为视频插值任务的定量评估指标。高 PSNR 和 SSIM 分数和低 IE 分数的方法在视频插值中更受青睐。我们使用箭头来表示偏好的值，即 PSNR ↑，SSIM ↑ 和 IE ↓。

使用合成事件训练数据。对于合成数据，我们从 GoPro、Adobe240 和 Vimeo90k 的训练集中采集视频序列。GoPro 数据集包含 22 个用于训练的不同视频和 11 个用于测试的额外视频，而 Adobe240 数据集有

112 个用于训练的序列和 8 个用于测试的序列。两个视频数据集都是用 GoPro 相机记录的，帧率为 240 fps，分辨率为 1280×720。Vimeo90k 数据集有 51312 个用于训练的三元组，其中每个三元组包含 3 个连续的视频帧，分辨率为 448×256 像素。所有收集到的视频序列都与使用流行的视频到事件模拟器 ESIM 生成的相应合成事件配对。

实验对比：在几个数据集上的定量比较如下：

Table 2. Quantitative comparison on Adobe240, GoPro, Middlebury (other) and Vimeo90k (interpolation) dataset with synthetic events.

Method	Frame	Event	Supervision	Adobe240						GoPro					
				7 skip (whole)			7 skip (center)			7 skip (whole)			7 skip (center)		
				PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓
E2VID [30]	✘	✔	✔	10.40	0.570	75.21	10.32	0.573	76.01	9.74	0.549	79.49	9.88	0.569	80.08
SepConv [23]	✔	✘	✔	32.31	0.930	7.59	31.07	0.912	8.78	29.81	0.913	8.87	28.12	0.887	10.78
QVI [38]	✔	✘	✔	32.87	0.939	6.93	31.89	0.925	7.57	31.39	0.931	7.09	29.84	0.911	8.57
DAIN [2]	✔	✘	✔	32.08	0.928	7.51	30.31	0.908	8.94	30.92	0.901	8.60	28.82	0.863	10.71
SuperSloMo [9]	✔	✘	✔	31.05	0.921	8.19	29.49	0.900	9.68	29.54	0.880	9.36	27.63	0.840	11.47
Time Lens [36]	✔	✔	✔	35.47	0.954	5.92	34.83	0.949	6.53	34.81	0.959	5.19	34.45	0.951	5.42
UnSuperSloMo [31]	✔	✘	✘	29.92	0.908	9.10	29.36	0.898	9.85	28.23	0.861	10.35	27.32	0.836	11.67
Ours	✔	✔	✘	34.14	0.950	6.25	33.22	0.942	6.64	34.02	0.960	5.02	33.39	0.952	5.59

Method	Frame	Event	Supervision	Middlebury (other)						Vimeo90k (interpolation)					
				3 skip			1 skip			3 skip			1 skip		
				PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓
E2VID [30]	✘	✔	✔	11.26	0.427	69.73	11.82	0.403	70.15	-	-	-	10.08	0.395	79.89
SepConv [23]	✔	✘	✔	25.51	0.824	6.74	30.16	0.904	3.93	-	-	-	33.80	0.959	3.15
QVI [38]	✔	✘	✔	26.31	0.827	6.58	31.02	0.908	3.78	-	-	-	-	-	-
DAIN [2]	✔	✘	✔	26.67	0.838	6.17	30.87	0.899	4.86	-	-	-	34.20	0.962	3.03
SuperSloMo [9]	✔	✘	✔	26.14	0.825	6.33	29.75	0.887	4.65	-	-	-	32.93	0.948	3.50
Time Lens [36]	✔	✔	✔	32.13	0.908	4.07	33.27	0.929	3.17	-	-	-	36.31	0.962	2.38
UnSuperSloMo [31]	✔	✘	✘	24.86	0.789	7.40	28.27	0.873	4.98	-	-	-	30.38	0.930	3.77
Ours	✔	✔	✘	30.91	0.887	4.89	32.74	0.912	3.63	-	-	-	35.12	0.963	2.79

由表二可以看出仅使用事件数据 E2VID 的方法性能最差，因为事件数据稀疏且仅包含相对强度变化，不足以重建高质量的中间帧。在事件数据的帮助下，同时使用强度图像和事件数据作为输入的方法(Time Lens)和本文提出的方法比仅使用强度图像作为输入的方法(SuperSloMo、DAIN、SepConv 和 QVI)表现更好。该篇文章是按照无

监督方法训练的甚至比大多数有监督的方法表现得更好。这可以归因于它在通过充分利用几乎连续捕获的事件流来精确建模非线性运动方面的强大能力。如图五只有 Time Lens 和该文的方法能够准确地模拟运动，才能预测一个合理的中间帧，特别是在车牌上。其他方法受到来自两个输入帧的估计不一致的影响，从而在数字上产生伪影。这表明，事件数据确实可以帮助建模复杂的运动与高质量的中间帧插值。

由于模拟事件没有背景噪声和有限的读出带宽，与真实事件存在差距，于是进行了在真实事件上训练数据，结果如下：

Table 3. Quantitative comparison on HQF and HS-ERGB dataset with real-world events.

Method	Frame	Event	Supervision	HQF				HS-ERGB (far)				HS-ERGB (close)			
				3 skip		1 skip		7 skip		5 skip		7 skip		5 skip	
				PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
E2VID [30]	✗	✓	✓	6.70	0.315	6.70	0.315	7.01	0.372	7.05	0.374	7.68	0.427	7.73	0.432
DAIN [2]	✓	✗	✓	26.10	0.782	29.82	0.875	27.13	0.748	27.92	0.780	28.50	0.801	29.03	0.807
SuperSloMo [9]	✓	✗	✓	25.54	0.761	28.76	0.861	24.16	0.692	25.66	0.727	27.27	0.775	28.35	0.788
RRIN [13]	✓	✗	✓	26.11	0.778	29.76	0.874	23.73	0.703	25.26	0.738	27.46	0.800	28.69	0.813
BMBC [25]	✓	✗	✓	26.32	0.781	29.96	0.875	24.14	0.710	25.62	0.742	27.99	0.808	29.22	0.820
Time Lens [36]	✓	✓	✓	30.57	0.900	32.49	0.927	32.31	0.869	33.13	0.877	31.68	0.835	32.19	0.839
UnSuperSloMo [31]	✓	✗	✗	23.47	0.740	26.11	0.852	23.80	0.651	25.81	0.700	26.72	0.732	28.38	0.741
Ours	✓	✓	✗	28.82	0.866	31.07	0.931	30.07	0.834	31.98	0.861	29.83	0.816	31.21	0.818

由图六看出没有事件输入的方法会产生许多伪影，尤其是在边缘。当使用仅帧插值方法重建帧序列时，可以发现帧中的单词受到影响，同时借助事件输入(Time Lens 和我们的方法)避免了伪影。

收获进展：

该文提出的方法的一个重要优点是它的无监督性质。以前的方法以监督的方式生成高质量的地面真值框架来训练 VFI 模型。它们要么需要像 HQF 那样 精心控制的慢拍，要么需要高速摄像机与事件摄像机之间复杂的配准，这是低成本不易获得的。相比之下，一种无监督的方法可以从任意视频和配对事件中受益，这可以很容易地通过 DAVIS 传感器大规模获得。大规模的无监督训练数据可以进一步提高 VFI 的性能。为了证明无监督训练的好处，我们选择大规模 DDD-17 数据集进行无监督微调，该数据集包含 346x260 像素 DAVIS 传感器的超过 12h 的记录。由于缺乏高速中间帧，以前的方法无法利用 DAVIS 传感器产生的低速帧和配对事件。由于我们的 TimeReplayer 算法依赖于无监督循环一致训练，DDD-17 数据集可以用于无监督微调。结果总结于表 5。

Table 5. Quantitative comparison on HQF dataset with real events.

Method	3 skip		1 skip	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Time Lens-syn	28.98	0.873	30.57	0.903
Time Lens-real	30.57(+1.59)	0.900(+0.027)	32.49(+1.92)	0.927(+0.024)
Ours	28.82	0.866	31.07	0.931
Ours+DDD17	31.54(+2.72)	0.920(+0.054)	33.93(+2.86)	0.934(+0.003)

为了获得更好的性能，Time Lens 需要对高成本的真实事件数据进行微调。相比之下，该文方法可以使用低成本的无监督事件数据进行微调，由于 DDD17 中的数据丰富，这超过了 Time Lens。

感悟：本文中所提出的 TimeReplayer 算法进行事件相机视频插值取得了很好的效果，在合成数据集上的表现优于先前算法，但是在实际数据集上的训练性能稍逊色，这可能源于实际数据中的噪声等干扰，可在后续研究中改善。