PEANUT: Prompt-Enhanced Ablation with Optical Flow-Based Neural Unit for Spatio-Temporal Consistency and VSR++ Clarity from $2D \rightarrow 3D$ Field

1. New Dataset

首先，除了一些常用于视频消除的官方数据集（如：DAVIS：https://openaccess.thecvf.com/content_cvpr_2016/papers/Perazzi_A_Benchmark_Dataset_CVPR_2016_paper.pdf 和 Youtube-VOS：https://arxiv.org/pdf/1809.03327)。另外，我们还自创收集了一些新的数据（新的测试集），以检测实际复杂应用场景、以及复杂指令下的消除（因为我们通过文字语义识别进行对象消除，所以我们需要运用更复杂、分辨率以及帧数更高的视频数据）。

传统数据痛点：

- 帧数太少

【可视化新老数据对比帧数可视化（尤其是 youtube 跨帧），每一帧的动态截图拼起来】

- 分辨率太低

【新数据统计】

- 画面较为简单，运动复杂性、场景变化性不足，以及可供语义理解的对象太少，并不能充分发挥本模型的作用
- 目标较为单一，视频中画面对象太少，不能起到很好的检测模型“混淆”判断能力

但是，Youtube 对希望处理复杂、长时序、多对象、多类别、现实场景（wild，in-the-wild）研究者来说，它可能不够“现实/多样”。 训练（training）3,471 视频 YouTube-VOS Dense（每隔约 5 帧，即 ~6 fps）标注；共约 5,945–6,459 个独立物体实例（unique object instances)。标注方式为 稠密标注（dense annotation)：训练集中每隔约 5 帧（≈6 fps）提供 segmentation mask / ground truth。鉴于你当前项目希望：

根据我们的任务： prompt 生成 mask → 对象消除（inpainting）→ 修复超分辨率 + 去噪（enhancement）处理可能存在 多对象 /复杂背景 /运动 /遮挡 /再现 /长序列 的视频。所以在模型训练上，我们用 YouTube- VOS 的大规模、多样性、多类别、多实例 +真实世界（in-the-wild）场景非常适合用作训练。但是在训练集方面，为了体现我们模型的性能，我们进一步收集了能够体现复杂性的数据：

我们基于以下原则去挑选新的额外测试数据。基于两个原则：

（1） Prompt 指引要正确！由于严格的 prompt 指令，所以 prompt 正确描述物体对象的行为以及修饰。【可视化不同 prompt 对于同一张图不同的定位】

（2） 基于该检验原则：生成视频掩码要正确！正确掩码才能正确地引导基于光流引导消除的 E2FGVI 模块。所以，我们在初步挑选数据时，首先要保证掩码要正确！所以我们运用掩码检测器去对采样视频备选进行过滤，确保挑选对象满足验证模型的以下能力：

一、语言能力：逼迫模型“听懂你在说谁”

同类多目标消歧（最能体现“语言理解”）能力： 通过属性/位置/动作来精准锁定对象，而不是“随便删一个”。

- 数据设计：场景里有多个相似目标

【可视化：相似目标的画面即可 sample 一定数量的 frames（5 张）】

二、时序能力：体现“视频级别”的优势，而不是单帧修图
复杂运动 & 相机运动恢复能力： 长序列跟踪 + 时序一致的修复，而不是一帧一帧乱补。
物体在大幅度跑动，
【可视化：大幅度跑动的目标 sample 帧】

三、背景重建能力：展示“修得干不干净、细不细腻”
复杂纹理 & 规则结构修复能力： 把被挡住的背景补得细致、结构保持一致。
【可视化，遮挡对象的视频帧率】

四、前景/背景和主题能力：展示模型能够很好地理解“什么是主体？主体和背景、前景地区分”
【头文字 D 效果】

2. P-MASk

首先，我们提出了一个对于掩码生成的任务：“Referring Video Object Segmentation (RVOS)”，给定一段视频 + 一个自然语言描述（referring expression，比如 “穿红衣服的那个人” / “左边那辆红车”），要求模型在视频中标出并追踪所指对象 — 输出每一帧该对象 segmentation mask。对于 RVOS 任务，过去的技术主要有：分割处理短片段 (clip-based) + 离线 (offline) 全视频处理，但这些都易丢失 “全局上下文 (global context)” 信息（例如对象跨 clip 出现、遮挡情况），以及正在运动时对象的修复。于是，我们提出：即 支持流式 (streaming) 视频处理 + 同时利用历史帧上下文 + 支持自然语言 (text) 提示 (prompts)，来通过人类自然语言的输入，来达到对视频中目标对象的逐帧掩码生成。
传统的基于 transformer 对视频中目标进行分割和跟踪，将“自然语言”作为查询(query)，并通过 Transformer 对视频中的目标进行分割和跟踪。这种方法利用了条件查询（conditional queries）和动态卷积核（dynamic convolution kernels）来处理视频帧，从而实现端到端的分割与跟踪【论文：https://openaccess.thecvf.com/content/CVPR2022/papers/Wu_Language_As_Queries_for_Referring_Video_Object_Segmentation_CVPR_2022_paper.pdf?】。 但是，这种方法容易丢失长时的上下文信息关联。所以，我们选择借鉴今年新推出的学术成果：(CVPR 2025) 并进行改进：https://openaccess.thecvf.com/content/CVPR2025/papers/Cuttano_SAMWISE_Infusing_Wisdom_in_SAM2_for_Text-Driven_Video_Segmentation_CVPR_2025_paper.pdf 提出的“基于 Meta 提出来的 SAM 架构【https://ai.meta.com/sam2/】+【https://arxiv.org/pdf/2408.00714】，引入多模态信息融合模块 + 校正模块”，来确保轻量化的部署（因为这个只是作为全流程的一个 pre-processing 预处理，所以应该追求轻量化部署！），以及侧重于实时和流式的视频处理，还有长时间的记忆联系。

基于轻量化部署，我们选择了 SAMWISE 模块，该模块去产生根据语义产生的视频对象掩码。

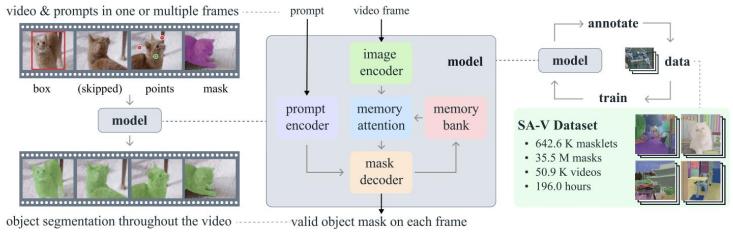


Figure 1 We introduce the Segment Anything Model 2 (SAM 2), towards solving the promptable visual segmentation task (a) with our foundation model (b), trained on our large-scale SA-V dataset collected through our data engine (c). SAM 2 is capable of interactively segmenting regions through prompts (clicks, boxes, or masks) on one or multiple video frames by utilizing a streaming memory that stores previous prompts and predictions.

SAM2 模型框架：https://arxiv.org/pdf/2408.00714

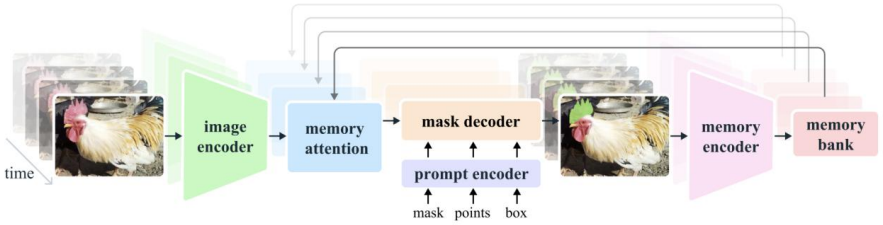


Figure 3 The SAM 2 architecture. For a given frame, the segmentation prediction is conditioned on the current prompt *and/or* on previously observed memories. Videos are processed in a *streaming* fashion with frames being consumed one at a time by the image encoder, and cross-attended to memories of the target object from previous frames. The mask decoder, which optionally also takes input prompts, predicts the segmentation mask for that frame. Finally, a memory encoder transforms the prediction and image encoder embeddings (not shown in the figure) for use in future frames.

SAM2 模型框架：https://arxiv.org/pdf/2408.00714

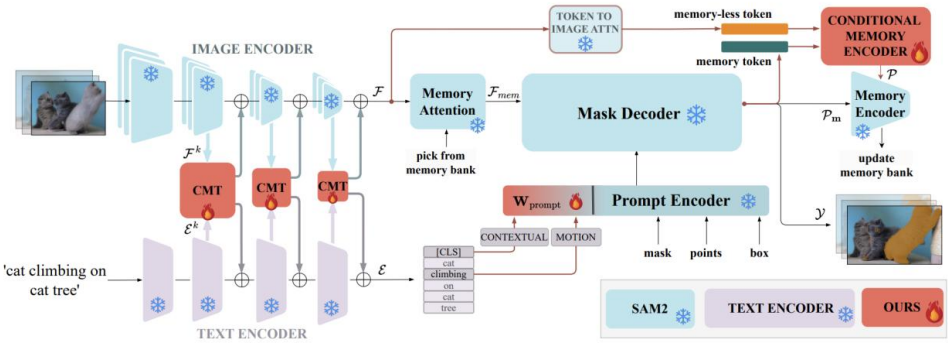


Figure 2. **Overview of SAMWISE.** We build on a frozen SAM2 and a frozen Text Encoder to segment images in video given a textual description. We incorporate the Cross-Modal Temporal Adapter (CMT) into the text and visual encoders at every intermediate layer k to model temporal dynamics within visual features while contaminating each modality with the other. Then, we extract the [CLS] and verb embeddings, namely Contextual and Motion prompts, from the adapted textual features and project them through a learnable MLP. The final embedding is used to prompt the Mask Decoder, which outputs the segmentation mask. Finally, the Conditional Memory Encoder detects when a new candidate object, aligned with the caption, appears in the frame, enabling SAM2 to dynamically refocus its tracking.

SAMWISE 框架：

https://openaccess.thecvf.com/content/CVPR2025/papers/Cuttano_SAMWISE_Infusing_Wisdom_in_SAM2_for_Text-Driven_Video_Segmentation_CVPR_2025_paper.pdf

我们在不改变 SAM2 原有权重（frozen weights），而通过 “adapter + memory correction” 的方式注入 语言（text）+ temporal（时间）+ multi-modal（多模态）能力。

具体来说，SAMWISE 的主要设计可以拆解为以下两个核心模块：

2.1 Cross- Modal Temporal Adapter Block (CMT)：融合语言+时序+视觉特征

目的：让模型在进行视频 segmentation 时，能够理解自然语言提示（text prompt），并且能够感知对象随时间（帧间）的变化（motion, temporal cues）。CMT 会接收三个模

态 (modalities) 的输入特征 — 视觉 (visual)、语言 (text)、以及历史帧记忆 (memory / temporal) 特征 — 然后通过一种 cross-modal + temporal-aware attention /融合机制，将这个跨膜态的信息进行融合。

2.2 Conditional Memory Encoder (CME) —— 修正 / 纠正 “tracking bias”

什么是 “tracking bias”？我们观察到，当使用 SAM2 + memory propagation 时，若目标对象在视频开始帧不出现 (或遮挡) — 模型可能一开始 “锁定 (track)” 到一个误对象 (distractor)；即使正确对象随后出现，也不会自动切换，因为 memory 依赖的是之前 mask propagation。(so, 目标对象出现在画面中的先后顺序很重要)。这种 “误判” 的现象在动态、复杂的视频 (有遮挡、对象交替出现、多个相似对象时) 尤为严重。

CME 的作用 (memory-free)：在每一帧，除了使用带 memory 的 features (memory- aware features) 得到 mask 之外，还生成一次 “memory-free (unbiased)” 的预测 (即不考虑过去追踪 history，仅根据当前 frame + text prompt 判断哪个对象最符合) — 这样可以判断当前是否有新的对象更符合语言描述 (caption)。如果“memory-free” 输出与 “memory-aware” 输出差异很大 (即可能是新对象)，CME 会 “纠正 (correct)” memory，使模型切换追踪目标。换句话说，CME 为 SAMWISE 提供一种动态纠错 (dynamic-correction) 的机制，使其在对象出现、消失、遮挡、重新出现等复杂情况下更加健壮。

因此，我们就得到了一个基于 prompt 的视频掩码帧生成。在不修改基础模型权重、仅通过轻量 adapter + memory- correction 机制 的条件下，把一个强大的视觉分割 + tracking 基础模型 (SAM2),转变为一个 支持自然语言 (text-driven)、支持流式视频 (streaming)、具有 temporal coherence 和语义理解能力的 RVOS 系统。这一部分只是作为数据预处理，根据人类输入语义，来生成视频的掩码帧数据。

3. NOF-Eraser

这是一个端到端可训练的框架，用于基于光流引导的视频修复（video inpainting）。具体来说，设计了一个名为 E2FGVI（End- to- End Flow- Guided Video Inpainting）的方法，通过三个模块（流完成、特征传播、内容幻觉 / 补充）联合训练，从而替换传统“先估计光流 → 像素传播 → 图像修补”那种手工拼接、流程分离的方式。

前面，在 2D 世界中，虽然我们运用 SAMWISE 的方法，根据语义提取到了图像目标对象的掩码，并且进行 LaMa 修复得到了不错的效果，但是从 2D -> 3D 存在一系列问题需要考虑：如果我们对视频中每一帧都进行 LaMa 修复的话（即使我们抽样提取关键帧来逐步计算）仍存在计算效率和计算时间的问题。另外，LaMa 只是针对单独一帧的图像进行修复，只依赖于单独目前该帧的信息，并不能在时间和空间上，和前面+后面的帧进行紧密的联系。所以这种方法无论从效率，还是质量上，都是不理想的。
总的来说，相较于图像修复（image inpainting），视频修复要同时考虑 空间结构一致性 和时间连贯性，因此难度更大。

经过论文的学习，我们找到了“流基方法”，关注帧间的信息。

3.1 流基方法

3.1.1 光流：

光流（optical flow）提供了视频帧之间的运动信息，是实现时间一致性的一个关键要素。利用光流引导传播可以更好地保留时间一致性。文章指出：“Among them, typical flow-based methods … consider video inpainting as a pixel propagation problem to naturally preserve the temporal coherence.”【https://arxiv.org/pdf/2204.02663】。算法认为视频中缺失的像素（即需要修复的区域）不应该完全由生成器（Generator）凭空生成，而应该通过从邻近帧中的已知信息沿着运动轨迹进行“借用”或“传递”，并且这种传播是通过 光流 (Optical Flow) 或更先进的 特征流 (Feature Flow) 来指导的。基于流的方法通过强制特征的传播遵循观测到的运动，确保了运动一致性和时间平滑性，因此，我们需要预测光流从而得知视频信息特征的传播,从而更好地联系帧间的信息以及更好的连接。Later, 我将下载 Optical Flow SDK from the NVIDIA developer zone（下面是从开发者博客截取的数据，要 cite 一下）.

Cite from: https://www.edge-ai-vision.com/2019/03/an-introduction-to-the-nvidia-optical-flow-sdk/

3.1.2 光流估计：

光流估计 (optical flow estimation)” 的目标是：从连续图像 (video / image sequence) 的强度 (intensity) 变化中，估计出图像中每个像素 (或区域) 对应的 2D 运动矢量 (velocity) —— 近似反映场景中物体或摄像机运动所产生的视觉变化。
【https://www.cs.toronto.edu/~fleet/research/Papers/flowChapter05.pdf】这一任务很重要，因为：运动是视觉系统的重要信号，光流可用于许多视觉任务 (3D 重建、运动分析、目标跟踪、分割、场景理解等），在计算机视觉与图像/视频处理领域，可靠的光流估计是基础模块。基础的经典方法有：

- 基础版：基于梯度方法：亮度恒常性 (Brightness constancy) 假设 & 梯度约束 (Gradient constraint) + 最小二乘 (Least- Squares, LS) 方法
【https://www.cs.toronto.edu/~fleet/research/Papers/flowChapter05.pdf】
- 改进：迭代 + 多尺度 (Iterative & Coarse-to-Fine Refinement), Robust Estimation (对抗 outliers / 异常), Motion Models (参数模型 / parametric)。。。等等很多
【https://www.cs.toronto.edu/~fleet/research/Papers/flowChapter05.pdf】

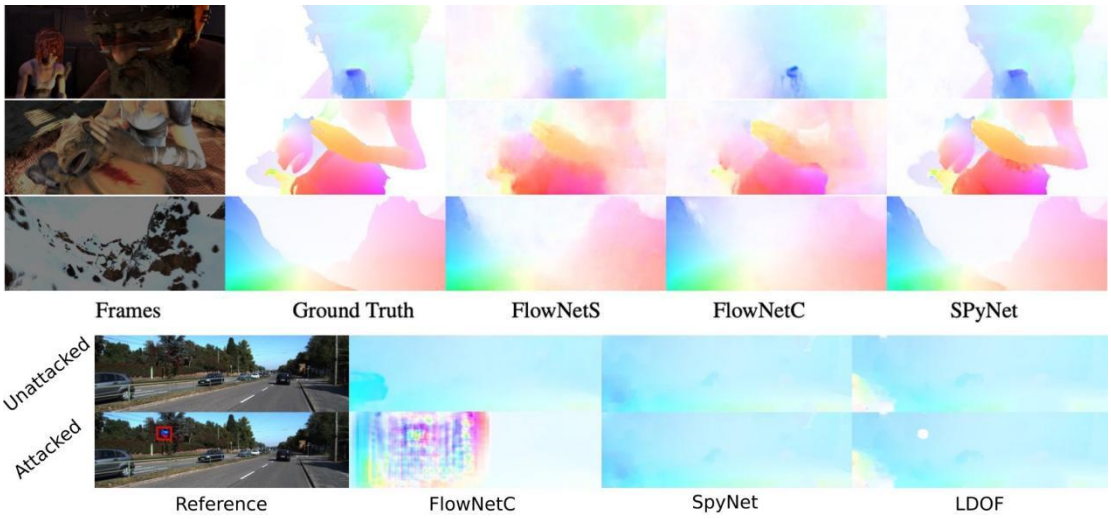
早期的光流预测经典方法如 Farneback 或 Lucas-Kanade，不足以应对复杂场景的快速移动以及变化剧烈的情况，所以我们运用基于深度学习框架的 SPyNet 来进行，它通过深度卷积

网络或 Transformer 来预测更精确的稠密光流场
【Farneback 方法存在多项式拟合假设：假设局部邻域内图像可以由一个二次多项式较好拟合 — 对于纹理非常复杂/高频/非平滑/强结构区域，并且如果帧间运动太大，邻域变形严重，多项式拟合 + 简单平移模型不够应对：
https://link.springer.com/chapter/10.1007/3-540-45103-X_50?】
【Lucas-Kanade 方法假设“小位移 + 局部恒定运动”—— 对于快速运动 / 大位移 / 复杂变形 / 遮挡 (occlusion) 等情况，容易失败：
https://www.sci.utah.edu/~gerig/CS6320-S2015/Materials/CS6320-CV-S2015-OpticalFlow-I.pdf?】

对于多种方法，我们得到了对不同算法进行光流估计任务的评估：

Algorithm	Accuracy	Speed (FPS)	Computational Requirements
Lucas-Kanade	Moderate	High	Low
Horn-Schunck	High	Low	High
FlowNet	High	Moderate	Moderate
LiteFlowNet	Very High	Moderate	Moderate
PWC-Net	Very High	High	High

Cite from: https://viso.ai/deep-learning/optical-flow/

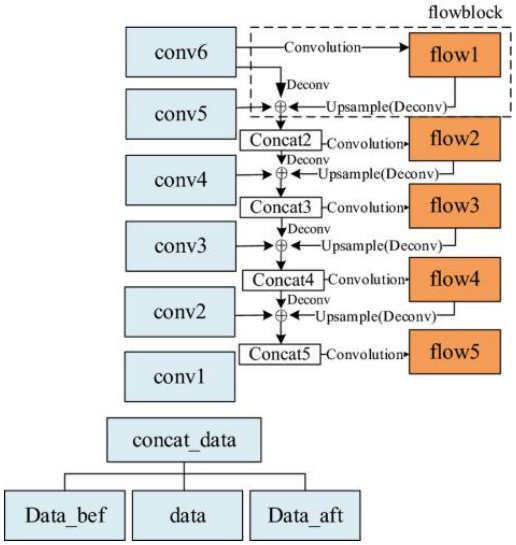


Cite from: https://is.mpg.de/ps/projects/learning-optical-flow
这张图展示了多种光流估计方法（FlowNetS、FlowNetC、SPyNet、LDOF）在不同攻击下的性能对比，其中“Ground Truth”代表真实的光流，其他方法在不同场景下的表现显示了各自的优势和局限性。

这个项目，基于多种效率和质量的平衡，我们将使用 SPyNet (Spatial Pyramid Network for Optical Flow Estimation) 来进行光流的预测。

3.1.3 SpyNet

SPyNet 的核心思想是将经典“空间金字塔（spatial pyramid）+coarse-to-fine（粗到细）”策略与深度学习（CNN）结合，用于光流（optical flow）估计 — 试图兼顾“传统光流方法（pyramid + warping）+ 深度学习（CNN）”的融合思路。
【https://openaccess.thecvf.com/content_cvpr_2017/papers/Ranjan_Optical_Flow_Estimation_CVPR_2017_paper.pdf?】



Cite from: the Flowchart of SPyNet architecture
https://www.researchgate.net/figure/Flowchart-of-SPyNet-architecture_fig2_349163951

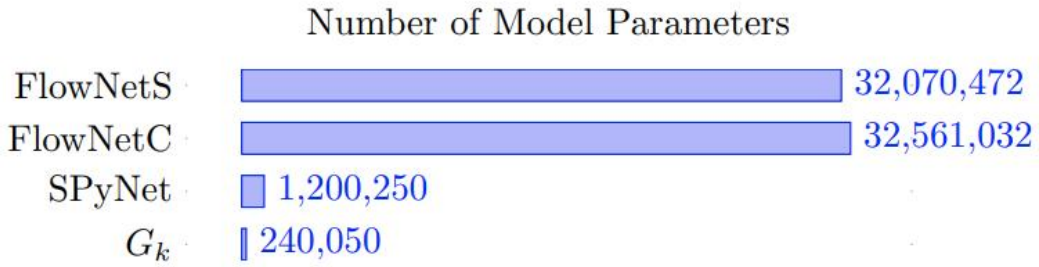


Figure 5. Model size of various methods. Our model is 96% smaller than FlowNet.

Cite from 体现 SpyNet 轻量化，利于轻量部署，节省计算资源。（要强调这个，因为我们的计算资源不多，所以才选这个效果和成本折中的框架，不然就纯用深度学习的框架了）
https://openaccess.thecvf.com/content_cvpr_2017/papers/Ranjan_Optical_Flow_Estimation_CVPR_2017_paper.pdf?

把光流（motion）作为 pipeline 一部分（例如结合帧间运动判断运动目标和跟踪，以及时间一致性 temporal coherence 等），SPyNet 是一个资源开销小、可嵌入（embedded / mobile）

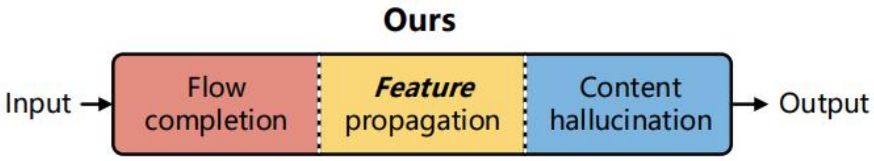
的选择 — 对于大规模 /批处理 /低资源系统尤其合适。可以把 SPyNet 作为 motion estimation 的基础 (baseline)，与静态图像分析 (object detection / segmentation) 结合 — 例如以 flow + appearance (图像特征) 联合判断目标移动 /变化 /遮挡。【https://openaccess.thecvf.com/content_cvpr_2017/papers/Ranjan_Optical_Flow_Estimation_CVPR_2017_paper.pdf?】

3.1.3 基于流的视频处理 (Flow - based video processing)

总括：作者强调，光流在多个视频任务中是一个强有力的先验 (先知约束提升时间信息对齐物理约束：光流必须遵循物体运动轨迹)，比如视频目标检测和定位、等等。 其中，视频恢复借助光流实现对齐或信息传播。需要聚合多帧低分辨率信息来重建高分辨率细节。光流用于将邻近帧的特征扭曲 (Warp) 到参考帧，确保聚合的像素是对应的，从而避免重影 (Ghosting) 效应。视频帧插帧 (VFI) 运动估计与补偿 模型首先估计两帧之间的光流，然后利用这个流来插值出中间帧的像素位置和颜色，实现运动补偿，确保生成帧运动自然平滑。

3.2 Main Process

过程综述：E2FGVI (End- to- End Flow- Guided Video Inpainting) 方法，通过三个模块 (流完成、特征传播、内容幻觉 / 补充) 联合训练，从而替换传统“先估计光流 → 像素传播 → 图像修补”那种手工拼接、流程分离的方式。



- 流基 https://arxiv.org/pdf/2204.02663

首先，模型利用一个编码器，将每帧映射到低分辨率空间 (堆叠 stacking 步长大于 1 的普通卷积层，逐渐将原本的二维分辨率降低，同时增加通道数，得到低分辨率特征表示)。紧接着，进行三大模块的逐步推进：流填充模块，特征传播模块，以及最后的内容幻觉模块。

- 流填充模块：对相邻帧完成光流估计 (前向、后向)
- 特征传播：利用第一步已经估计好的光流，在特征层中执行双向传播 & 融合
- 内容幻觉：采用多层“时空焦点 Transformer”，结合传播后的特征与来自非本地帧的特征，以生成最终补齐的帧特征。

最后一个帧解码器 (frame - level decoder) 将补齐后的特征上采样回原始分辨率，输出修复后的视频。总的框架图如下所示：https://arxiv.org/pdf/2204.02663

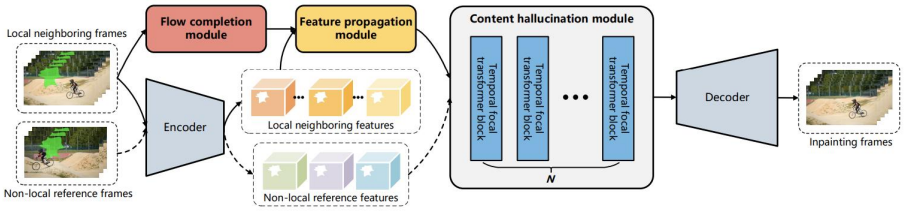


Figure 2. Overview of the proposed End-to-End framework for Flow-Guided Video Inpainting (E^2FGVI). It consists of 1) a frame-level content encoder, 2) a flow completion module, 3) a feature propagation module, 4) a content hallucination module which is composed of multiple temporal focal transformer blocks, and 5) a frame-level decoder.

E2FGVI 框架图 https://arxiv.org/pdf/2204.02663

E2FGVI 全流程解释

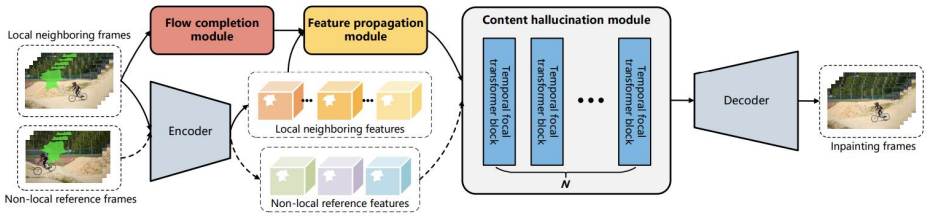


Figure 2. Overview of the proposed End-to-End framework for Flow-Guided Video Inpainting (E^{2}FGVI). It consists of 1) a frame-level content encoder, 2) a flow completion module, 3) a feature propagation module, 4) a content hallucination module which is composed of multiple temporal focal transformer blocks, and 5) a frame-level decoder.

- 解释图解：基于新提出的新模块“Content Hallucination Module”，作者需要一开始就采集对于时间 t 下的两重数据，一个是当前时间步下的帧图像(local neighboring frames)，另一个是跨越未来时间步 t+N 下的帧图像（Non-local reference frames：后面会用到）.

对于时间步 t：

- 首先将当前时间下的帧信息，传入到“流填充模块”，该模块训练一个光流估计网络，学习跨时间的光流位移预测。该网络可以逼近双向真实光流（Ft→t+1 ,Ft→t-1），所以这一步是用来估计在某个时间时间步下，往前 / 往后 物体对象 光流移动估计

- 特征传播模块（Feature Propagation Module）：核心 DCN 提取得到深层特征信息 $\hat{E}_b^t$

  - 构建 DCN 的前置条件！！！：

当前帧简单编码信息（Local neighbouring features E_t）+ 简单卷积层估计得到两个东西：预测一个偏移量 $\Delta(F_{t\to t+1})$ + 对应权重掩码 $(W_{t\to t+1})$

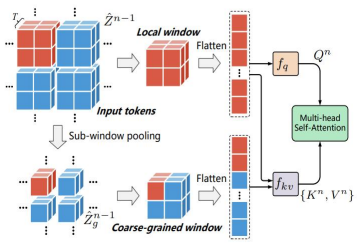
$$[W_{t\to t+1}, \Delta F_{t\to t+1}] = C_b(E_t; W(\hat{E}_{t+1}^b, \hat{F}_{t\to t+1}), \hat{F}_{t\to t+1})$$

- need: $[W_{t\to t+1}, \Delta F_{t\to t+1}]$

the forward flow $\hat{F}_{t\to t+1}$. The warped feature can be further merged with current content feature E^t and updated through a backward propagation function $\mathcal{P}_b(\cdot)$:

$$\hat{E}_b^t = \mathcal{P}_b(E^t, \mathcal{W}(\hat{E}_b^{t+1}, \hat{F}_{t\to t+1})), \quad (3)$$

- Content Hallucination Module：需要结合 local 和 non-local 的信息（均为低分辨率多通道的特征表示，but local 是高级特征表示，而 non-local 是低级特征表示），通过 soft-split (SS) 方法来对这两种特征进行融合（本质：先将两种初始特征，一高级一低级，来首先直接通道拼接，然后再进行 soft-split 的 overlapped patch embedding on the concatenated local and non-local temporal features:）！具体来说，SS 操作就是对拼接后的特征划分为多个小的补丁，并对这些补丁进行 embeddings 处理，允许不同补丁之间有重叠部分。切完之后再进行多头自注意力计算... 融合计算体现：在 K^n 中，包含{K_l, K_g, local 和 grained 两种}



Suppose T_{nl} is the number of selected non-local frames. $\mathbf{E}_{nl} \in \mathbb{R}^{T_{nl}\times\frac{H}{4}\times\frac{W}{4}\times C}$ is the encoded features of all non-local neighbors. $\hat{\mathbf{E}}_l \in \mathbb{R}^{T_l\times\frac{H}{4}\times\frac{W}{4}\times C}$ is the local temporal feature through concatenating the results in Eq. (6) at the temporal dimension. We use a soft split operation [33] to perform overlapped patch embedding on the concatenated local and non-local temporal features:

$$Z^0 = \mathrm{SS}([\hat{\mathbf{E}}_l, \mathbf{E}_{nl}]) \in \mathbb{R}^{(T_l+T_{nl})\times M\times N\times C_e}, \quad (7)$$

where SS denotes the operation of soft split. Z^0 is the embedded token that contains both local and non-local temporal information. $M \times N$ is the embedded spatial dimension, and C_e is the feature dimension.

This operation can be processed in parallel. We concatenate corresponding keys and values respectively by $K^n = \{K_l^n, K_g^n\}$ and $V^n = \{V_l^n, V_g^n\}$, and then calculate the local self-attention for Q_l^i:

$$\mathrm{Attention}(Q^n, K^n, V^n) = \mathrm{Softmax}\left(\frac{Q^n (K^n)^T}{\sqrt{C_e}}\right) V^n. \quad (9)$$

Note that the attention function also can work in a multi-head manner. An example is shown in Fig. 4.

Finally, the whole process in the n-th focal transformer block is formulated as

$$Z'^n = \mathrm{MFSA}(\mathrm{LN}_1(Z^{n-1})) + Z^{n-1}, \quad (10)$$

$$Z^n = \mathrm{F3N}(\mathrm{LN}_2(Z'^n)) + Z'^n, \quad (11)$$

where MFSA and LN denote the multi-head focal self-attention and layer normalization [1], respectively. We use F3N [33] to link the connections across embedded patches.

3.2.1 流填充模块 (Flow Completion)

- 原始视频帧 => 输入帧下采样，降低分辨率 => 初始化光流估计网络 (F) & 初始光流传播位移 => 学习网络，逼近双向真实光流 ($F_{t\to t+1}$, $F_{t\to t-1}$) 【光流用 F_t 来表示时间】

- 本项目先将原始被遮掩的视频帧下采样压缩至 (1/4) 分辨率，记为 ($X_{t\downarrow}$)，

$$\text{noted as } X^t_{\downarrow} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 3}.$$

- 使用一个流估计网络 (F) 得到初始前向流 ($F_{i\to j} = F(X_{i\downarrow}, X_{j\downarrow})$)。

- 通过训练使其输出接近 “真实光流” ($F_{t\to t+1}$, $F_{t\to t-1}$)（使用被遮掩区外的真实帧估计） 的 L(* 1)L(* 1) 损失：

$$\mathcal{L}_{flow} = \sum_{t=1}^{T-1} \|\hat{F}_{t\to t+1} - F_{t\to t+1}\|_1 + \sum_{t=2}^{T} \|\hat{F}_{t\to t-1} - F_{t\to t-1}\|_1, \tag{2}$$

where $F_{t\to t+1}$ and $F_{t\to t-1}$ are the ground truth forward and backward flow, respectively, which are calculated from original uncorrupted videos.

与传统方法相比，作者指出其流完成模块一次性前馈即可完成，而不是传统多阶段初始化＋细化。

3.2.2 特征传播 (Feature Propagation)

特征传播的目的：将传统基于光流的特征传播 => 基于 可变形卷积 的鲁棒对齐

3.2.21 Traditional Optical Flow Wrapping

传统 wrapping 传播模块信息聚合 + 对齐（基于光流扭曲特征）

从上下文编码器得到局部邻近帧的特征集合 (Et∈RH/4×W/4×C | t=1…Tl)

当前 + 未来信息的融合：有效的利用起来两个信息：一个是当前时刻 t，另一个是下一个时刻 t+1.

当前 t 时刻：拥有当前时刻简单编码(通过编码器 encode 当前信息)特征 Et，然后利用下一个时刻的、下一时刻已聚合未来多步时间帧信息（way：通过“传播模块”的操作，是一种特殊的特征传播策略，具有迭代信息的作用）的 E_{t+1} ：

- E_{t+1} 在其自身时间步（即 t+1）的传播模块中，已经聚合了来自 t+2,… t+3,… 未来若干帧信息的特征

- 对相邻帧（或已修复帧）特征进行 扭曲 (Warping) 操作，将信息对齐到当前帧（通过扭曲操作，将相邻帧的信息 & 当前帧对齐）。通过 双向传播 (Bidirectional Propagation)（向前传播和向后传播）来更好地处理运动遮挡 (Disocclusion) 问题

- 从而，得到了 Eˆtb 的计算表达式，该表达式融合了当前时刻 t 的信息 + 未来多个时间步骤的信息。
- 对于 Eˆtb 的计算式子如下：

从上下文编码器得到局部邻近帧的特征集合 $(E_t \in \mathbb{R}^{H/4\times W/4\times C} \mid t = 1 \ldots T_l)$。
以 $(\hat{F}_{t\to t+1})$ 为例：它从帧 (t) 指向帧 (t+1) 的运动。作者将 (E_{t+1}) 的“向后传播特征” $(\hat{E}^b_{t+1})$ 通过 warping（基于光流）至当前时刻，然后与 (E_t) 融合：
$[\hat{E}^b_t = P_b(E_t,; W(\hat{E}^b_{t+1},, \hat{F}_{t+1\to t}))]$
其中 (W(·)) 表示使用光流进行空间 warping。

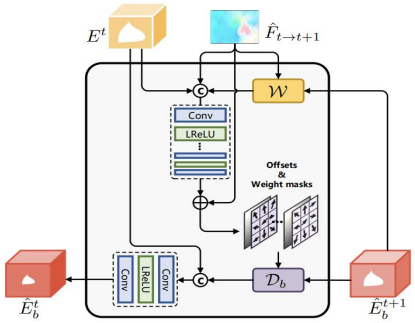


Figure 3. An example of using the completed forward flow $\hat{F}_{t\to t+1}$ to guide the feature backward propagation, where $\oplus$ and © denote an addition operation and a concatenation operation, respectively. Note that the backward flow will act in the opposite direction.

- 计算得到 Eˆtb 的可视化，体现上面 $[\hat{E}^b_t = P_b(E_t,; W(\hat{E}^b_{t+1},, \hat{F}_{t+1\to t}))]$ 的计算过程（这个是基础的流基方法推导特征双向传播的计算公式... 下面有更加新的作者用到的方法）

3.2.22 DCN based Optical Flow Wrapping

DCN 替代传统 Warping 的核心：模型不再直接使用光流去扭曲特征，而是利用光流（Ft）去指导 DCN 采样参数（预测 DCN 的参数？）。可变形卷积（modulated deformable convolution）来增强鲁棒性，

the forward flow $\hat{F}_{t\to t+1}$. The warped feature can be further merged with current content feature E^t and updated through a backward propagation function $\mathcal{P}_b(\cdot)$:

$$\hat{E}^t_b = \mathcal{P}_b(E^t, \mathcal{W}(\hat{E}^{t+1}_b, \hat{F}_{t\to t+1})), \quad (3)$$

where $\mathcal{W}(\cdot)$ denotes the spatial warping operation based on optical flow, $\hat{E}^t_b$ is the backward propagation feature at the t-th time step, and the propagation function $\mathcal{P}_b(\cdot)$ represents two convolutional layers with a LeakyReLU [37] activation.

DCN-Based Step1：利用轻量卷积网络（C_b），预测 DCN 输入

- 具体而言，他首先预测一个偏移量 $\Delta(F_{t\to t+1})$ + 对应权重掩码 $(W_{t\to t+1})$ 作为可变形卷积的采样参数：
  $[W_{t\to t+1}, \Delta F_{t\to t+1}] = C_b(E_t,; W(\hat{E}^b_{t+1}, \hat{F}_{t\to t+1}), \hat{F}_{t\to t+1})$
- need: $[W_{t\to t+1}, \Delta F_{t\to t+1}]$
- by 轻量化卷积层

input: >>> 时间 t 时刻下：

- 普通特征E_t （普通的编码器得到）
- warping对齐得到的$\hat{E}^b_t$ + 光流信息 $\hat{F}_{t+1\to t}$

output: <<< acted as DCN's input:

- 偏移量（对DCN可变卷积采样点的二维修正量） $\Delta F_{t\to t+1}$
- 对应每个采样点的贡献程度 $W_{t\to t+1}$
- 总公式：
  轻量网络输出得到DCN-params 采样参数:

$$C_b(E_t,; W(\hat{E}^b_{t+1}, F_{t\to t+1}),; F_{t\to t+1}) == output ==> [W_{t\to t+1}, \Delta F_{t\to t+1}]$$

DCN-Based Step2（DCN based Optical Flow Wrapping）：执行可变形卷积操作
接着用该偏移量 + 权重掩码在特征层执行可变形卷积，替代传统 warping。
操作公式：（普通卷积的版本上，添加一些可变、自适应的要素）

- p_0: 采样操作的起始中心点 - 默认是(0,0)
- p_n: 卷积核的固定偏置 - 假设3x3卷积，那么 p_n 就是: (0,0), (0,1), (0,2) ... (2,2)
- so, $p_0 + p_1$ 才是最原始采样的地方

可变卷积

$$Y(p_0) = \sum_{p_n \in \mathcal{R}} W(p_n) \cdot X(p_0 + p_n + \Delta p_n) \cdot m_n$$

- 在传统卷积上，增添了 "二维偏移量" + "掩码权重"
- Δp_n: 采样点在时间下的位移偏移 ΔF_{t→t+1}
- m_n: 每个采样点的相对重要性-每个采样点的贡献强度（Modulated部分）。这可以过滤掉对齐不良的像素（如运动边界、遮挡区域）的信息

DCN-Based Step3（DCN based Optical Flow Wrapping）：最终特征融合

最终，对前向传播特征$(\hat{E}_t^f)$与后向 $(\hat{E}_t^b)$ 用一个 (1×1) 卷积融合：

$$[\hat{E}_t = I(\hat{E}_t^f, \hat{E}_t^b)]$$

其中 (I) 是 (1×1) 卷积。

该设计的优点：

> 特征层传播允许更大的感受野、可学习的传播路径，对光流误差的敏感性减弱。
> 结合可变形卷积使得模型在“如果光流估计有误”时仍有一定柔性补偿。
> 流估计 + 特征传播组成的模块均可训练，从而避免了传统流程中“估计 → 固定 → 传播”的割裂。

3.2.23 时空焦点 Transformer（Temporal Focal Transformer）

作者指出，仅靠局部邻近帧传播可能不足，因为有些缺失内容的来源可能是非邻近帧（non- local frames）中的信息。（例如被遮挡物重新出现在更早 / 更远帧中）因此，他们选取 T_{n1} 个非本地参考帧的编码特征 $E_{n1} \in \mathrm{R}^{T_{n1} \times H/4 \times W/4 \times C}$，以及局部传播后的特征（$\hat{E}_1 \in \mathrm{R}^{T1 \times H/4 \times W/4 \times C}$）。然后做一个 soft- split 操作（类似于 patch embedding）将它们拼接为 token 序列：

$$[Z_0 = SS([\hat{E}_1, E_{n1}]) \in \mathrm{R}^{(T1+Tn1) \times M \times N \times Ce}]$$

其中 (M×N) 是空间 patch 数量，(C_e) 是嵌入维度。

> 相比标准 Vision Transformer 的全局 self- attention，作者采用了 **Focal Transformer 机制**（原用于 2D 图像）并将其扩展到 **3D 时空窗口**（st × sh × sw）形式。其核心思想是：

- 在局部子窗口中执行细粒度 attention（fine - grained local attention）。
- 在粗粒度子窗口中执行粗略 attention（coarse global attention）以捕获长程依赖。
- 3D 格式下，对于 focal transformer 的切割方法（类似于二维的 patch，但是 3D 多了时间维度）

from 2D to 3D. Specifically, we first split the input token Z^{n-1}, where $n \in [1, N]$ and N is the stacking number of focal transformer blocks, into a grid of sub-windows with size $s_t \times s_h \times s_w$. The split token $\hat{Z}^{n-1} \in \mathbb{R}^{(\frac{(T_l+T_{nl})}{s_t} \times \frac{M}{s_h} \times \frac{N}{s_w} \times C_e) \times (s_t \times s_h \times s_w)}$ can be directly used for computing fine-grained local attentions. To perform global attention at the coarse granularity, a linear embedding layer f_p is used to pool the sub-windows spatially via $\hat{Z}_g^{n-1} = f_p(\hat{Z}^{n-1}) \in \mathbb{R}^{(\frac{(T_l+T_{nl})}{s_t} \times \frac{M}{s_h} \times \frac{N}{s_w} \times C_e) \times s_t}$. We then calculate the query, key, and value through two linear projection layers f_q, f_{kv}:

$$Q^n = f_q(\hat{Z}^{n-1}), \quad \{K_l^n, K_g^n, V_l^n, V_g^n\} = f_{kv}(\{\hat{Z}^{n-1}, \hat{Z}_g^{n-1}\}). \tag{8}$$

【可以画图可视化去理解！！！】

- 数学形式：
  - 切分 token 为子窗口 ($\hat{Z}^{n-1} \in \mathbb{R}^{(...)\times(st\times sh\times sw)\times C_e}$)。
  - 通过线性 embedding 层 (f_p) 池化得到粗粒度 ($\hat{Z}_g^{n-1}$)。
  - 计算 query ($Q^n = f_q(\hat{Z}^{n-1})$)；同时 ($K_l^n, K_g^n, V_l^n, V_g^n = f_{kv}(...)$)。
  - Attention 计算形式为：

$$[\text{Attention}(Q^n, K^n, V^n) = \text{Softmax}\Big(\frac{Q^n (K^n)^T}{\sqrt{C_e}}\Big) V^n]$$

其中 ($K^n = K_l^n, K_g^n, V^n = V_l^n, V_g^n$)。

  - 整体 Transformer 块：

$$[Z_n^0 = \text{MFSA}(\text{LN}_1(Z^{n-1})) + Z^{n-1}]$$

$$[Z^n = \text{FFN}(\text{LN}_2(Z_n^0)) + Z_n^0]$$

其中 MFSA 表示多头焦点自注意力、LN 是归一化、FFN 是前馈网络。

- 通过这种方式，模型既能从最近帧中获取及时信息，也能从更远帧中"检索"完整的内容，从而提升修复质量和时间一致性。

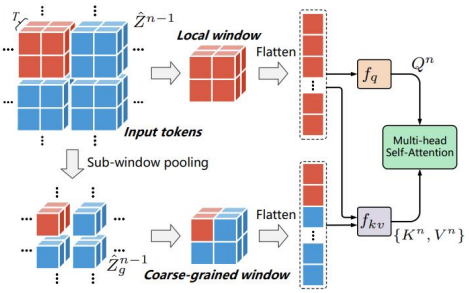


Figure 4. Illustration of temporal focal self-attention. Here we use the window size of $2 \times 2 \times 2$ as an example. We can see that the keys and values $\{K^n, V^n\}$ contain both fine-grained local information and coarse-grained global information.

这个 E2FGVI 基于端到端训练的框架，成功整合了流估计、传播、生成整合。在生成阶段引入时空焦点 Transformer，能更好地融合本地/非本地时空信息，并且在效率上有显著提升（模型轻量化、易部署 + 推理速度快）。

但是缺点也很明显：

对大运动场景或遮掩面积非常大的情况仍表现欠佳。【可视化展现】

对于非典型场景，例如复杂遮挡、摄像机快速移动（or 跳帧快转），变化迅猛剧烈的情况适应性较差。【可视化展现】

训练数据集的局限性：训练数据的特性约束了模型学习的能力，因为运用 Youtube-VOS 和 DAVIS 数据集，都是中低水平的运动方式，以及画面信息量较少，目标对象少，so 对于一些复杂场景的修复效果一般般。

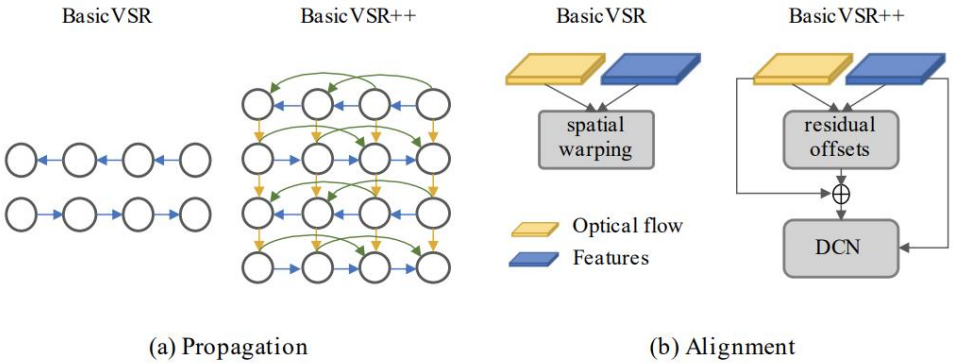
4.UR-Net

上面已经做完的端到端流基图像修复工作，我们需要对修复水平进行超分辨重建。

考虑到整个流程的轻量化部署，我们选用了轻量但保证修复效果的 BasicVSR++。该方法在保持较低计算复杂度的同时，大幅提高了视频超分辨率的恢复性能。该模型不仅在视频超分辨率任务中表现出色，还能有效地推广到其他视频恢复任务，如压缩视频增强。【https://arxiv.org/pdf/2104.13371】。

通过增强的传播和对齐机制,使得视频超分辨率网络能在复杂视频场景中保持一致性和更高的质量。 本工作是基于基本的 BasicVSR 来进行改进：

- 原 BasicVSR: 采用简单的双向传播（如下图），这种严重限制了信息的交互和特征聚合。所以新的 BasicVSR 通过运用“二阶网格传播”，引入更多的依赖关系，尤其是在当前帧信息有限的情况下，能够从其他帧获取有用的补充信息。



基础 BasicVSR 和改进后的 BasicVSR++信息流向对比图

工作流程：

(1) 输入数据：输入通过 E2FGVI 模块处理后的低分辨率视频帧
(2) 特征提取：
① 空间特征提取：首先，使用卷积网络从每个视频帧中提取空间特征
② 时间特征提取：接着，模型使用时间卷积（或光流信息）来提取帧之间的时序特征
(3) 残差学习与时空信息融合：BasicVSR++ 使用残差学习方法，通过输入的低分辨率视频帧和网络学习到的残差特征进行融合。网络通过这种方式学习到如何恢复高分辨率的视频细节。

(4) 生成高分辨率输出：模型通过结合时空特征和残差信息，输出每一帧的高分辨率版本。为了保持时间一致性和空间一致性，BasicVSR++ 在推理过程中还利用多尺度特征融合技术

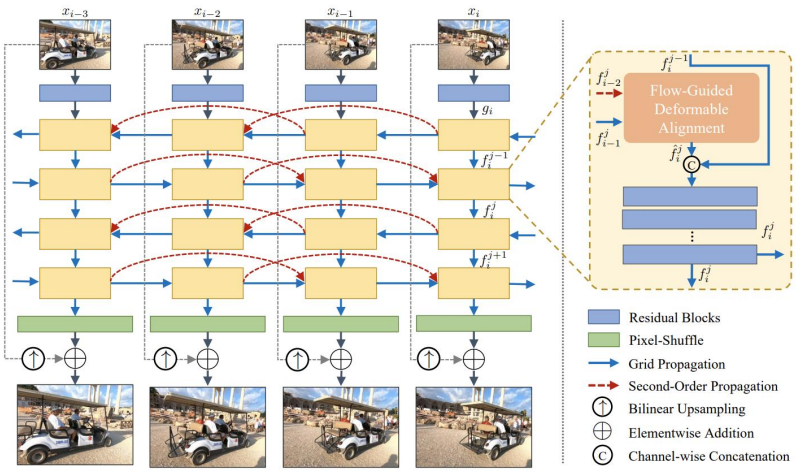


Figure 2: **An Overview of BasicVSR++.** BasicVSR++ consists of two modifications to improve propagation and alignment. For propagation, we introduce second-order propagation (blue solid lines) to refine features bidirectionally. In addition, second-order connection (red dotted lines) is adopted to improve the robustness of propagation. Within each propagation branch, flow-guided deformable alignment is proposed to increase the offset diversity while overcoming the offset overflow problem.

- BasicVSR++ 模型框架图(https://arxiv.org/pdf/2104.13371)

BasicVSR++ 通过时空卷积网络和光流信息，能够高效地捕捉视频中的时空关系。这使得模型在恢复视频细节的同时，也能够保证视频帧之间的一致性，减少了时间模糊和失真。

引入光流信息的设计，使得 BasicVSR++ 能够更好地捕捉帧之间的运动，避免了传统视频超分辨率方法中常见的运动模糊问题