使用扩散模型进行分布外推的视频预测

摘要

视频预测作为计算机视觉中的关键任务，旨在基于当前帧序列准确预测未来帧。然而，由于未来的不确定性和复杂的时空动态，视频预测面临着巨大的挑战。本文在复现现有的 ExtDM（外推分布扩散模型，cvpr2024）方法的基础上，提出了一项改进：在其时空注意力机制中引入空间位置注意力。通过在原有的时空窗口注意力（Spatiotemporal Window Attention, STW Attention）中增加空间位置信息，我们的优化方法能够更有效地捕捉视频帧中的空间结构，从而提升特征表达能力和预测准确性。为了验证所提出方法的有效性，我们在 KTH 动作数据集上进行了实验。实验结果表明，改进后的模型在峰值信噪比（PSNR）和结构相似性（SSIM）等指标上显著优于原始的 ExtDM 模型，尤其在处理复杂动态场景时表现更加稳定。此外，消融实验进一步验证了空间位置注意力对提升模型性能的关键作用，同时保持了计算效率在可接受范围内。本研究不仅验证了空间位置注意力在视频预测任务中的有效性，也为未来的视频预测模型优化提供了新的思路。通过结合时空动态与空间结构信息，我们的方法展示了在视频预测领域中取得更高精度和一致性的潜力。

关键词：视频预测；扩散模型；时空注意力

1 引言

在当今信息化时代，视频内容在我们的生活中无处不在，从社交媒体到安全监控，再到自动教师汽车。视频预测技术，即使用机器学习算法预测未来视频帧的内容，是计算机视觉领域的一个重要研究方向。这一技术可以帮助系统理解视频内容的动态变化，预测未来时间，从而做出更加只能的决策。尽管现有技术已经能够处理简短视频序列的预测问题，但在长期视频预测，计算效率和预测精确性方面仍面临挑战。

《ExtDM：Distribution Extrapolation Diffusion Model for Video Prediction》[42] 一文中提出的 ExtDM 模型，利用扩散过程和分层分布适配器结合 3D U-Net 架构的方法，对视频帧进行长期预测。该模型的创新之处在于它如何处理时间动态和复杂场景的建模，显示出优于其他方法的潜力。然而，尽管原始模型已经取得了一定的成功，但对于如何进一步提高其效率和准确性，尤其是在不同的视频数据集上的表现，仍有探索的空间。

通过复现并改进 ExtDM 模型，不仅可以深入理解模型的内部机制和优缺点，还可以探索新的方法来提升视频预测技术的性能。这对于提高视频监控系统的预警能力、优化自动驾驶车辆的决策过程、以及丰富视频编辑和动画生成工具的功能等应用都具有重要意义。

2 相关工作

视频预测可预测未来帧，以增强检测和分割等应用程序。最近的进展包括视频扩散模型，它将高斯噪声转换为详细的视频预测。尽管有这些改进，但准确模拟远程时间动力学在计算上仍然具有挑战性。

2.1 视频预测

视频预测能够在像素级别 [2,4,5,10,36,39,41] 预测未来的帧，并模拟帧之间的变化 [8,18,26,28,30,33,40]，这对下游应用如表示 [6,16,32,35]、检测 [21–23]、分割 [7,14,15,44] 和恢复 [25,47–49] 至关重要。在早期的工作中，一些研究 [9,20] 提出了基于随机变分推断的方法，这些方法能够显示的提取空间和时间信息。例如，PRNN [34] 构建了时空 LSTM，SLAMP [1] 从外观和光流中学习先验分布，而 MOSO [29] 则将帧分解为运动，场景和对象张量。

2.2 视频扩散模型

视频扩散模型学习将高斯分布噪声转换为与视频相关的分布 [3,24,27,38]，这一过程依赖于多轮条件引导的去噪迭代。VDM [12] 首次证明了扩散模型完成视频任务的可行性。RVD [37] 提出一个扩散模型，每次预测下一个视频帧的残差误差。MCVD [31] 基于 2D 卷积通过压缩维度实现了一个通用的多输入多输出视频扩散模型。RAMViD [13] 引入了随机掩码，并构建了一个 3D 卷积视频扩散模型。LVDM [11] 使用 3D 自编码器和分层机制在潜在控价生成任意长度的视频。

2.3 其他预测模型

随着深入研究 [19,35,43,45,46]，各种方法都在努力预测未来，但由于每种解决方案的固有缺陷，模拟长期时间动态仍然具有挑战性。直接方法带来高计算成本，因此难以在低资源设备上部署。在上下文中学习的方法依赖于从当前帧推断出的语义线索，这与未来的帧存在差距。

3 本文方法

3.1 本文方法概述

ExtDM 所提出的流程图如图1所示，给定一系列条件帧 x_c,ExtDM 的目标是通过充分利用外观和运动线索来预测视频中的未来帧 x_p, 设 x_p, x_c 的长度分别为 u, v。论文中的方式流程可以概述为三个部分：(i) 运动自编码器 (第 3.1 节)，(ii) 分布外推扩散模型 (第 3.2 节)，以及 (iii) 运动自编码器重建 (第 3.1 节)。运动自编码器的编码器将条件帧 x_c 投影为一系列运动线索 m_c(即光流和遮挡图)。然后，分层分布适配器通过一对高斯过程将特征外推到未来。时空窗 (STW)U-Net 通过注意力机制以未来特征为参考，生成未来线索 $\hat{m}_p$。最后，运动自编码器的解码器从预测的运动线索 $\hat{m}_p$ 和条件帧 x_c 重建未来帧 x_p。

为了为预测未来结果打下基础，论文连理了一个双射编码，包含两个映射函数：$x_c -> m_c, \hat{m}_p -> x_p$

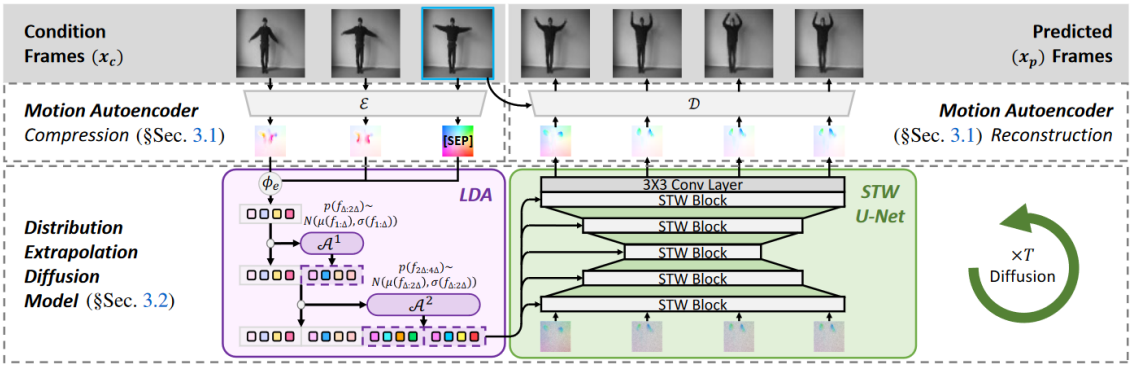


图 1. ExtDM 由三个主要组件组成：运动自动编码器通过压缩和重建在像素空间和运动空间之间构建双射变换。分层分布适配器将未来帧的特征推断为从条件帧导出的移位分布。此外，构建的 STW U-Net 以外推特征为指导，在时空维度之间进行稀疏和跨步关注，以鼓励特征交互。

3.2 运动自编码器

采用一个轻量级的运动自编码器对视频进行压缩，该自编码器在运动线索与视频帧之间进行双摄变换。基于轻量级自编码器架构 [17]，运动自编码器由两个阶段组成：编码器 E 从帧中提取运动线索，解码器 D 从运动线索中重构视频帧。

3.2.1 运动自编码器压缩

为了从一系列条件帧中提取运动线索，编码器 $\varepsilon(\cdot,\cdot)$ 以成对的方式估计视频帧之间的光流和遮挡图。对于条件帧 $x_c = \{x_i \in RCHW | i = 1, ..., u\}$，长度为 u, 我们提取每个条件帧 x_i 与最后一个条件帧 x_u(用于重建的关键帧) 之间的运动线索。这些成对的帧被送入编码器，以估计它们之间的运动相关性，包括光流 w_i 及其相应的遮挡图 o_i。

$$m_c = \left\{ m_i \in \mathbb{R}^{3hw} \mid m_i = \mathcal{E}(x_i, x_u) = \begin{bmatrix} w_i \\ o_i \end{bmatrix} \right\} \tag{1}$$

3.2.2 运动自编码器重建

借助外推得到的未来运动线索 $\hat{m}_p = \{\hat{m}_j \in \mathbb{R}^{3hw} | j = 1, ..., v\}$ 和关键帧，解码器 $D(\cdot,\cdot)$ 以成对的方式重建未来帧，凡是与编码器相似，我们将关键帧与第 j 哥未来帧的预测运动线索配对。条件帧的潜在表示 z_u 首先在光流 w_j 的指导下进行变形。考虑到遮挡，变形后的表征通过结合每个预测未来帧的遮挡图 o_j 进一步过滤，具体为 $o_j \odot W(x_u, w_j)$。这一表征随后被送入网络 G 中，用于修补遮挡区域。这里，$W(z, w)$ 是在光流 w 的指导下对特征 z 进行的变形操作，而 $\odot$ 是元素级的乘积。最终哦该得到的重建帧。

$$x_p = \left\{ x_j \in \mathbb{R}^{3HW} \mid x_j = D(\hat{m}_j, x_u) = G(o_j \odot W(z_u, w_j)) \right\} \tag{2}$$

Algorithm 1 LDA: Pytorch-style Pseudocode

```
# f: input feature (N C T H W)
# phi_e: encoding layer
# phi_d: decoding layers
# L: number of layers

f = phi_e(f) # encoding condition frames
for l in range(L):
    r = f
    mu, var = est(f) # Gaussian prior
    f_h = (f - mu) / std
    mu = m_est(f_h) + mu
    var = (1 + v_est(f_h)) * var
    f_h = phi_d[l](f_h) # inferencing future frames
    f = f_h * var + mu
    f = torch.cat([r,f], dim=2)

# distribution estimation
def est(f, eps=1e-5):
    f_var = f.view(N, C, T, -1).var(dim=3) + eps
    f_std = f_var.sqrt().view(N, C, T)
    f_mean = f.view(N, C, T, -1).mean(dim=3)
    return f_mean, f_std
```

图 2. 算法一 LDA：pytorch 风格的伪代码

3.3 分布外推扩散模型

论文提出了一种分布外推扩散模型，通过一系列向后（去噪）步骤来外推运动线索 $\hat{m}_p$。基于高斯混合模型的假设，设计了分层分布适配器对未来特征的移动分布进行因果建模，并进一步引入时空注意力来融合外推特征和普通特征。

利用从潜在表示 z_c 中提取的运动线索 m_c 和外观特征，论文的视频扩散模型由前向函数 $\{q_t\}_{t\,[0,1]}$ 组成，用于将一系列噪声添加到未来帧 $m_p^1 \sim q_1(m_p^1)$ 中，和一个后向函数 $\{p_t\}_{t\in[0,1]}$，通过提出的时空窗口 U-Net 从高斯噪声 $p_1(m_p^0) := \mathcal{N}(0, \mathcal{I})$ 预测未来帧 $\epsilon_\theta(m_p^t, c)$。将外观特征（潜在表示 z_c）和运动特征（运动线索 m_c）作为指导 c。为了弥合现在和未来之间的差异，论文通过提出的分层分布适配器将条件框架 f_c 的指导外推到未来 f_p。

3.3.1 分层分配适配器

给定长度为 Δ 的条件帧的特征，LDA 旨在产生未来帧的外推特征。输入特征 f_c 首先被输入投影仪 Φ_e 以利用条件帧之间的时间相关性，然后以多层方式外推到未来。对于第 l 层，我们通过单层适配器 $\mathcal{A}(l)$ 从当前帧 $f_{1:\Delta}$ 预测未来帧 $\hat{f}_{\Delta:2^l\Delta}$，详细过程如图2

$$
\begin{aligned}
f_{1:\Delta} &= \phi_e(J_c), \\
\hat{f}_{1:2\Delta} &\triangleq (\hat{f}^{\Delta}_{1:2\Delta-1}, A^{(l)}(\hat{f}^{\Delta}_{1:2\Delta-1})), \\
f_p &= (\hat{f}_{1:\Delta}, \ldots, \hat{f}^{\Delta}_{2L-1:2\Delta}).
\end{aligned} \tag{3}
$$

3.3.2 时空窗 U-Net

采用由参数 θ 驱动的 3D-Uet$\epsilon_\theta(m_p^t, c)$ 作为去噪工具，处理视频序列中的噪声。该网络不仅继承了传统 3D-Unet 的上采样和下采样结构，还通过引入时空窗口 (STW)U-Net 增强了其功能。

针对来自 LDA 的指导特征 f_g 和待精化的特征 f_x，首先将时空特征分割成具有特定窗口大小的分区，并在至此那个下一轮 STW 注意力之前移动这些跟前，提高时空连贯性，交叉注意力估计为：

$$
f_{x\to g} = \text{softmax}\left(\frac{[T(f_x)W^Q][T(f_g)W^K]^T}{\sqrt{d}}\right) T(f_x)W^V \tag{4}
$$

并最大化函数和线性投影矩阵 W^Q, W^k, W^V, 其中 d 设定为特征的通道数，从而准确预测和重建未来帧中的运动线索。

4 复现细节

4.1 与已有开源代码对比

原论文的实验代码公布在 github 上，但是没有公布模型的训练参数。这里在源代码的基础上增添控制一个参数 $path$, 为 0 代表使用原论文中的方法，使用的是时间注意力。为 1 代表使用我自己的融合时间和空间的注意力，如图3

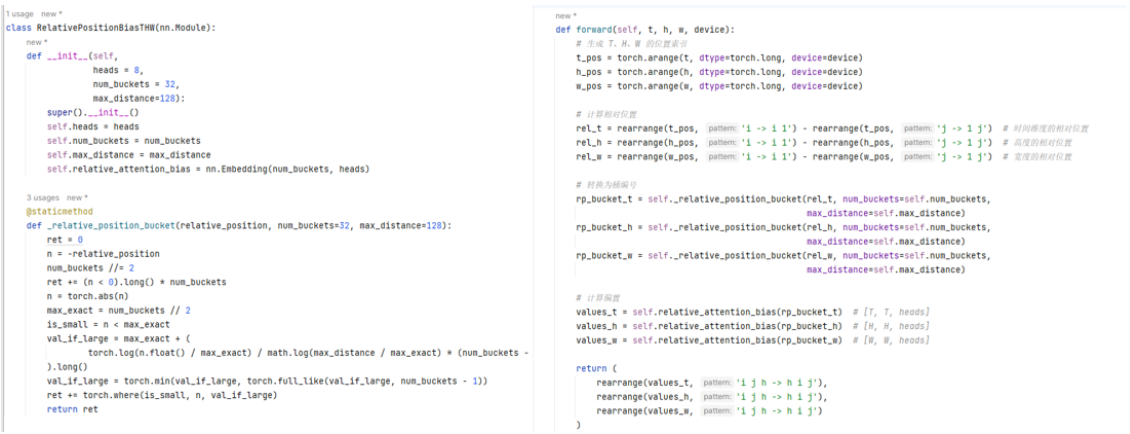
```
parser.add_argument( *name_or_flags: "--path", default=0,type=int,help="0 is use Original Time  Attention, "
                                                "1 is use combine Attention")
```

图 3. 添加控制参数

增加空间尺度上的偏差，然后与原来的时间位置偏差进行融合如图4，形成联合偏差，之后再次进行注意力的计算。其中，空间位置的偏差也是新增的类，具体代码如图5

```python
if path ==1:
    time_rel_pos_bias, height_rel_pos_bias, width_rel_pos_bias = self.rel_pos_bias_thw(tc + tp, h, w,
                                                                                       device=x.device)

    target_size = time_rel_pos_bias.shape[-1]  # 获取时间维度的大小

    height_rel_pos_bias_resized = F.interpolate(
        height_rel_pos_bias.unsqueeze(0), size=(target_size, target_size), mode="bilinear",
                                    align_corners=False
    ).squeeze(0)  # 插值后形状: [8, 30, 30]

    width_rel_pos_bias_resized = F.interpolate(
        width_rel_pos_bias.unsqueeze(0), size=(target_size, target_size), mode="bilinear",
                                    align_corners=False
    ).squeeze(0)
    height_rel_pos_bias_resized = height_rel_pos_bias_resized.unsqueeze(1).expand(-1, target_size, -1,
                                                                                  -1)  # [heads, T, T, T]
    width_rel_pos_bias_resized = width_rel_pos_bias_resized.unsqueeze(2).expand(-1, -1, target_size,
                                                                                -1)  # [heads, T, T, T]
    time_rel_pos_bias_expanded = time_rel_pos_bias.squeeze(0).unsqueeze(1).expand(-1, 30, 30, 30)  # [8, 30,


    # 融合
    alpha = self.alpha.view(-1, 1, 1, 1)  # [heads, 1, 1, 1]
    beta = self.beta.view(-1, 1, 1, 1)  # [heads, 1, 1, 1]

    combined_rel_pos_bias = (
            alpha * time_rel_pos_bias_expanded +
            beta * (height_rel_pos_bias_resized + width_rel_pos_bias_resized)
    )
```

图 4. 联合注意偏差



```python
1 usage  new *
class RelativePositionBiasTHW(nn.Module):
    new *
    def __init__(self,
                 heads = 8,
                 num_buckets = 32,
                 max_distance=128):
        super().__init__()
        self.heads = heads
        self.num_buckets = num_buckets
        self.max_distance = max_distance
        self.relative_attention_bias = nn.Embedding(num_buckets, heads)

    3 usages  new *
    @staticmethod
    def _relative_position_bucket(relative_position, num_buckets=32, max_distance=128):
        ret = 0
        n = -relative_position
        num_buckets //= 2
        ret += (n < 0).long() * num_buckets
        n = torch.abs(n)
        max_exact = num_buckets // 2
        is_small = n < max_exact
        val_if_large = max_exact + (
                torch.log(n.float() / max_exact) / math.log(max_distance / max_exact) * (num_buckets -
        ).long()
        val_if_large = torch.min(val_if_large, torch.full_like(val_if_large, num_buckets - 1))
        ret += torch.where(is_small, n, val_if_large)
        return ret
```

```python
    new *
    def forward(self, t, h, w, device):
        # 生成 T、H、W 的位置索引
        t_pos = torch.arange(t, dtype=torch.long, device=device)
        h_pos = torch.arange(h, dtype=torch.long, device=device)
        w_pos = torch.arange(w, dtype=torch.long, device=device)

        # 计算相对位置
        rel_t = rearrange(t_pos,  pattern: 'i -> i 1') - rearrange(t_pos,  pattern: 'j -> 1 j')  # 时间维度的相对位置
        rel_h = rearrange(h_pos,  pattern: 'i -> i 1') - rearrange(h_pos,  pattern: 'j -> 1 j')  # 高度的相对位置
        rel_w = rearrange(w_pos,  pattern: 'i -> i 1') - rearrange(w_pos,  pattern: 'j -> 1 j')  # 宽度的相对位置

        # 转换为桶编号
        rp_bucket_t = self._relative_position_bucket(rel_t, num_buckets=self.num_buckets,
                                                     max_distance=self.max_distance)
        rp_bucket_h = self._relative_position_bucket(rel_h, num_buckets=self.num_buckets,
                                                     max_distance=self.max_distance)
        rp_bucket_w = self._relative_position_bucket(rel_w, num_buckets=self.num_buckets,
                                                     max_distance=self.max_distance)

        # 计算偏置
        values_t = self.relative_attention_bias(rp_bucket_t)  # [T, T, heads]
        values_h = self.relative_attention_bias(rp_bucket_h)  # [H, H, heads]
        values_w = self.relative_attention_bias(rp_bucket_w)  # [W, W, heads]

        return (
            rearrange(values_t,  pattern: 'i j h -> h i j'),
            rearrange(values_h,  pattern: 'i j h -> h i j'),
            rearrange(values_w,  pattern: 'i j h -> h i j')
        )
```

图 5. 空间偏差

4.2 创新点

本实验的主要创新点在于引入了空间位置注意力机制，该机制是对传统时空注意力框架 STW U-Net 的一个重要扩展。此创新有效增强了模型在处理视频数据时的空间解析能力，能够更准确地捕捉并利用视频帧内的空间信息。通过这种改进，模型不仅能更敏感地响应场景中的重要空间特征，而且能显著提升预测未来帧的准确性。该机制通过对每个帧中不同空间位置的特征进行差异化处理，优化了特征的表达和信息的提取，使得整体的视频分析和处理更加精细和高效。这一改进为复杂动态场景的视频预测提供了一种新的解决策略，展示了在

保持高计算效率的同时，如何通过空间维度的深入挖掘来提升模型性能。

5 实验结果分析

模型使用两阶段进行训练，在第一阶段训练运动自编码器的编码器和解码器，第二阶段训练模型的分布外推扩散。

本实验未对第一阶段进行改进，采用源代码，在 KTH 数据集上进行训练的到的结果如下：

fvd	ssim	psnr	lpips
224.3	0.913	35.020	0.028

第二阶段是采用第一阶段训练的编码器和解码器并对其进行冻结，对原始论文的改进在第二阶段，仍在 KTH 数据集上进行实验，对比结果如下：

	fvd↓	ssim↑	psnr↑	lpips↓
原始模型	567.4	0.620	22.067	0.217
改进后	**553.9**	**0.635**	**22.565**	**0.213**

可以发现，改进后的各项指标均由于原始模型。

6 总结与展望

本研究在视频预测领域取得了一定的进展，尤其是在 KTH 动作数据集上实现了对未来 40 帧的有效预测，展示了引入空间位置注意力机制后模型在处理简单动作序列方面的潜力。然而，当应用于多人数据集和更复杂的彩色数据集时，模型表现出一定的局限性，预测时长仅有 2-3 帧。特别是在多人场景中，模型的扩展性和预测深度显著受限，第一阶段的训练便受到了限制，无法成功提取运动区域。此外，模型的庞大和训练周期长达数月的问题也严重制约了其实用性和进一步发展。

在未来的工作中，我们将重点放在优化和简化模型架构上，以减少计算资源消耗并加快训练速度，期望缩短训练周期同时保持或提升模型的预测精度。针对处理多人数据集的挑战，将探索更精细的特征提取技术和改进注意力机制，以增强模型对复杂交互和动态变化的捕捉能力，并考虑引入场景分割和个体识别技术以提升多人场景下的性能。此外，我们也将致力于开发新的算法或框架以支持更长时间序列的预测，并提高模型跨不同数据集的泛化能力，从而使模型能够适应更广泛的应用场景和更高复杂度的数据环境，推动视频预测技术向更高的准确性、效率和实用性方向发展。

参考文献

[1] Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel, and Luc Van Gool. Towards high resolution video generation with progressive growing of sliced wasserstein gans. *arXiv preprint arXiv:1810.02419*, 2018.

[2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[4] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7608–7617, 2019.

[5] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. Strpm: A spatiotemporal residual predictive model for high-resolution video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13946–13955, 2022.

[6] Xiuwen Chen, Li Fang, Long Ye, and Qin Zhang. Deep video harmonization by improving spatial-temporal consistency. *Machine Intelligence Research*, 21(1):46–54, 2024.

[7] Yadang Chen, Chuanyan Hao, Zhi-Xin Yang, and Enhua Wu. Fast target-aware learning for few-shot video object segmentation. *Science China Information Sciences*, 65(8):182104, 2022.

[8] Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Randomized conditional flow matching for video prediction. *arXiv preprint arXiv:2211.14575*, 2022.

[9] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International conference on machine learning*, pages 1174–1183. PMLR, 2018.

[10] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11474–11484, 2020.

[11] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2(3):4, 2022.

[12] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[13] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022.

[14] Duojun Huang, Xinyu Xiong, De-Jun Fan, Feng Gao, Xiao-Jian Wu, and Guanbin Li. Annotation-efficient polyp segmentation via active learning. *arXiv preprint arXiv:2403.14350*, 2024.

[15] Duojun Huang, Xinyu Xiong, Jie Ma, Jichang Li, Zequn Jie, Lin Ma, and Guanbin Li. Alignsam: Aligning segment anything model to open context via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3205–3215, 2024.

[16] Guoli Jia and Jufeng Yang. S 2-ver: Semi-supervised visual emotion recognition. In *European conference on computer vision*, pages 493–509. Springer, 2022.

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[18] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019.

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[20] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[21] Jiaming Li, Jiacheng Zhang, Jichang Li, Ge Li, Si Liu, Liang Lin, and Guanbin Li. Learning background prompts to discover implicit knowledge for open vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16678–16687, 2024.

[22] Xin Liu, Guobao Xiao, Riqing Chen, and Jiayi Ma. Pgfnet: Preference-guided filtering network for two-view correspondence learning. *IEEE Transactions on Image Processing*, 32:1367–1378, 2023.

[23] Xin Liu and Jufeng Yang. Progressive neighbor consistency mining for correspondence pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9527–9537, 2023.

[24] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. Vdt: An empirical study on video diffusion with transformers. *arXiv preprint arXiv:2305.13311*, 3(5):9, 2023.

[25] Siwei Ma, Li Zhang, Shiqi Wang, Chuanmin Jia, Shanshe Wang, Tiejun Huang, Feng Wu, and Wen Gao. Evolution of avs video coding standards: twenty years of innovation and development. *Science China Information Sciences*, 65(9):192101, 2022.

[26] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[27] Yaniv Nikankin, Niv Haim, and Michal Irani. Sinfusion: Training diffusion models on a single image or video. *arXiv preprint arXiv:2211.11743*, 2022.

[28] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[29] Mingzhen Sun, Weining Wang, Xinxin Zhu, and Jing Liu. Moso: Decomposing motion, scene and object for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18727–18737, 2023.

[30] Stephen Tian, Chelsea Finn, and Jiajun Wu. A control-centric benchmark for video prediction. *arXiv preprint arXiv:2304.13723*, 2023.

[31] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022.

[32] Lijuan Wang, Guoli Jia, Ning Jiang, Haiying Wu, and Jufeng Yang. Ease: Robust facial expression recognition via emotion ambiguity-sensitive cooperative networks. In *Proceedings of the 30th ACM international conference on multimedia*, pages 218–227, 2022.

[33] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International conference on learning representations*, 2018.

[34] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, S Yu Philip, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, 2022.

[35] Changsong Wen, Xin Zhang, Xingxu Yao, and Jufeng Yang. Ordinal label distribution learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23481–23491, 2023.

[36] Haixu Wu, Zhiyu Yao, Jianmin Wang, and Mingsheng Long. Motionrnn: A flexible model for video prediction with spacetime-varying motions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15435–15444, 2021.

[37] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 25(10):1469, 2023.

[38] Siyuan Yang, Lu Zhang, Yu Liu, Zhizhuo Jiang, and You He. Video diffusion models with local-global context guidance. *arXiv preprint arXiv:2306.02562*, 2023.

[39] Xi Ye and Guillaume-Alexandre Bilodeau. Vptr: Efficient transformers for video prediction. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3492–3499. IEEE, 2022.

[40] Xi Ye and Guillaume-Alexandre Bilodeau. A unified model for continuous conditional video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3603–3612, 2023.

[41] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations*, 2020.

[42] Zhicheng Zhang, Junyao Hu, Wentao Cheng, Danda Paudel, and Jufeng Yang. Extdm: Distribution extrapolation diffusion model for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19310–19320, 2024.

[43] Zhicheng Zhang, Pancheng Zhao, Eunil Park, and Jufeng Yang. Mart: Masked affective representation learning via masked temporal distribution distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12830–12840, 2024.

[44] Pancheng Zhao, Peng Xu, Pengda Qin, Deng-Ping Fan, Zhicheng Zhang, Guoli Jia, Bowen Zhou, and Jufeng Yang. Lake-red: Camouflaged images generation by latent background knowledge retrieval-augmented diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4092–4101, 2024.

[45] Sicheng Zhao, Guoli Jia, Jufeng Yang, Guiguang Ding, and Kurt Keutzer. Emotion recognition from multiple modalities: Fundamentals and methodologies. *IEEE Signal Processing Magazine*, 38(6):59–73, 2021.

[46] Sicheng Zhao, Xingxu Yao, Jufeng Yang, Guoli Jia, Guiguang Ding, Tat-Seng Chua, Bjoern W Schuller, and Kurt Keutzer. Affective image content analysis: Two decades review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6729–6751, 2021.

[47] Shihao Zhou, Duosheng Chen, Jinshan Pan, Jinglei Shi, and Jufeng Yang. Adapt or perish: Adaptive sparse transformer with attentive feature refinement for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2952–2963, 2024.

[48] Shihao Zhou, Mengxi Jiang, Shanshan Cai, and Yunqi Lei. Dc-gnet: Deep mesh relation capturing graph convolution network for 3d human shape reconstruction. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 171–180, 2021.

[49] Shihao Zhou, Mengxi Jiang, Qicong Wang, and Yunqi Lei. Towards locality similarity preserving to 3d human pose estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.