实时角色模拟与神经变形技术：基于 Houdini 与 UE5 的 ML Deformer 及 ML Groom 流程深度剖析

1. 引言：实时图形学中的非线性变形范式转移

在计算机图形学与实时渲染技术的演进历程中，角色变形（Character Deformation）始终是跨越“恐怖谷”效应的关键障碍。传统的线性混合蒙皮（Linear Blend Skinning, LBS）与双四元数蒙皮（Dual Quaternion Skinning, DQS）虽然计算效率极高，足以应对骨骼驱动的刚性运动，但在表现生物软组织（Soft Tissue）的复杂物理行为时却显得力不从心。肌肉的体积保持（Volume Preservation）、皮肤在筋膜上的滑动、脂肪的惯性抖动以及布料与毛发的高频褶皱，本质上属于非线性变形范畴，其物理规律难以通过简单的线性插值算法进行拟合。

随着硬件光线追踪能力的普及与张量核心（Tensor Cores）在消费级显卡上的集成，一种新的技术范式——机器学习变形器（Machine Learning Deformer）应运而生。该范式主张将高精度的离线物理模拟数据（Ground Truth）“蒸馏”进轻量级的神经网络中，从而在实时引擎（如 Unreal Engine 5）内以极低的推理开销重现电影级的变形效果。与此同时，针对高维数据（如毛发 Groom）的 ML 驱动流程，通过降维理论与神经网络的结合，解决了传统物理模拟在实时帧率下无法处理百万级自由度（Degrees of Freedom, DoF）的算力瓶颈。

本报告将深入剖析从 Houdini 到 UE5 的端到端技术链路，涵盖物理数据生成的拓扑逻辑、神经网络架构的数学原理、降维算法的理论博弈以及实时渲染管线中的性能优化策略，旨在为技术美术与图形工程师提供一份详尽的实施蓝图。

---

2. 物理模拟与数据生成：从离线解算到训练集构建

高质量的训练数据是机器学习模型泛化能力的基础。在 Houdini 中构建数据生成管线（Pipeline）时，核心挑战在于如何在解算精度、计算效率与数据的拓扑一致性之间找到最佳平衡点。

2.1 软组织解算器的演变：Vellum、FEM 与 Otis 的博弈

在生成肌肉与皮肤变形数据时，Houdini 提供了多种物理求解方案，每种方案在数学模型上存在本质差异，直接影响了生成数据的物理真实度与训练集构建的时间成本。

2.1.1 Vellum (PBD) 的稳定性优势

Vellum 求解器基于位置动力学（Position-Based Dynamics, PBD）或者是其扩展形式 XPBD（Extended PBD）。其核心思想并非直接求解牛顿第二定律中的力与加速度，而是通过直接修正粒子的位置来满足几何约束（Constraints）。

- 数学特性：PBD 系统通过迭代投影（Iterative Projection）来求解约束，这意味着它无条件稳定（Unconditionally Stable），即使在极大的时间步长下也不会“爆炸”。
- 在 **ML** 流程中的应用：由于其极高的稳定性和较快的解算速度，Vellum 常用于生成大量的布料和基础皮肤变形数据。通过构建四面体（Tetrahedral）约束，Vellum 能够模拟体积保持特性[1]。
- 局限性：PBD 是一种物理近似，它难以精确模拟基于材料属性（如杨氏模量、泊松比）的应力-应变关系。这导致生成的肌肉变形往往缺乏生物组织的“重量感”与非线性硬化特征[1]。

2.1.2 有限元方法 (FEM) 的物理精度

有限元方法（Finite Element Method）通过将连续介质离散化为有限个单元，利用连续介质力学（Continuum Mechanics）方程求解物体内部的应力张量。

- 数学特性：FEM 求解涉及构建和求解大型稀疏线性方程组（$Kx = f$），能够精确反映非线性弹性材料（如 Neo-Hookean 模型）的物理行为[1]。
- 数据质量：FEM 生成的数据在体积守恒和应力分布上具有极高的真实度，特别是对于肌肉的被动拉伸和主动收缩（Activation）表现最为准确。
- 计算瓶颈：FEM 的计算复杂度通常是 $O(n^2)$ 或更高，导致单帧解算时间极长，难以在有限时间内生成数万帧的训练数据[1]。

2.1.3 Otis Solver：GPU 加速的统一解算架构

Houdini 21 引入的 Otis Solver 代表了软组织模拟的最新方向。它是一种专为生物组织设计的 GPU 加速求解器，旨在结合 Vellum 的速度与 FEM 的精度[4]。

- 单步解算（**Single-Pass Architecture**）：传统的流程往往需要分层解算（骨骼 -> 肌肉 -> 筋膜 -> 脂肪 -> 皮肤），层级间的数据传递复杂且容易出错。Otis 允许在一个统一的求解器中同时模拟肌肉、脂肪和皮肤，极大地简化了约束管理（Constraint Management）[5]。
- 性能表现：利用 GPU 的并行计算能力，Otis 能够在保持 FEM 级物理精度的同时，将解算速度提升数倍，使其成为当前生成大规模 ML 训练集的首选方案[4]。

2.2 姿态空间采样理论 (Pose Space Sampling)

神经网络的学习效果高度依赖于训练样本在姿态空间（Pose Space）中的覆盖率。如果采样点过于稀疏或分布不均，模型在推理未见姿态（Unseen Poses）时会出现严重的伪影。

2.2.1 随机采样与高斯分布 (Gaussian Distribution)

最朴素的采样方法是在关节的旋转极限（Joint Limits）内进行均匀随机采样。然而，人体运动并非均匀分布，大多数动作集中在舒适区（Comfort Zone），而极限姿态出现的频率较低。Houdini 的 ml pose generate 节点采用了基于高斯分布的采样策略[7]。通过将采样的概率密度函数（PDF）集中在零位（Rest Pose）附近，同时保留长尾分布以覆盖极限角度，可以确保模型在常用姿态下具有最高的预测精度，而在极端姿态下也能保持鲁棒性。这种策略被称为“重要性采样”（Importance Sampling）在变形训练中的应用。

2.2.2 运动范围 (ROM) 与训练集清洗

为了进一步提高数据质量，技术美术通常会引入真实的运动捕捉数据（Motion Capture）作为种子，并在其基础上添加高斯噪声（Gaussian Noise）进行数据增强。这种方法生成的样本（Samples）被称为“运动范围”（Range of Motion, ROM）数据。

- 数据清洗：并非所有随机生成的姿态都是合法的。例如，头部穿入胸腔或手肘反向折叠的姿态必须在模拟前被剔除（Culling），否则这些“脏数据”会污染神经网络的权重，导致模型在推理时产生不可预测的震荡 [8]。

2.3 拓扑一致性与差值提取 (Delta Extraction)

神经网络预测的核心目标通常不是顶点的绝对世界坐标，而是相对于基础蒙皮（LBS）的偏移量（Deltas）。

$$\Delta v = v_{sim} - v_{LBS}$$

在 Houdini 中，这一过程通过 guide deform 和 shapediff 节点实现 [7]。

1. 逆向变形：将模拟后的高模（High-res Mesh）通过骨骼逆变换（Inverse Bone Transform）转换回静止姿态（Rest Pose）空间。
2. 差值计算：在静止空间下计算模拟顶点与 LBS 顶点的位置差。这种基于残差（Residual）的学习策略极大降低了神经网络需要拟合的数据动态范围（Dynamic Range），从而提升了收敛速度和最终精度 [10]。

---

3. 神经网络架构剖析：UE5 ML Deformer 的模型演进

Unreal Engine 5 的 ML Deformer 框架提供了多种模型架构，每种架构在推理解析度、显存占用和计算开销上各有优劣。理解这些架构的底层数学原理对于选择合适的生产流程至关重要。

3.1 Vertex Delta Model (VDM)：直接映射的尝试与局限

Vertex Delta Model 是最早期的尝试，其核心思想是构建一个全连接神经网络（Fully Connected Network, FCN），直接输入关节旋转，输出每个顶点的 (dx, dy, dz) 偏移量 [12]。

- 架构特点：
  - 输入层：关节旋转参数（通常为 6D 旋转向量或四元数）。
  - 隐藏层：若干层全连接层，使用 ELU 或 ReLU 激活函数。
  - 输出层：$N \times 3$ 个浮点数，对应 N 个顶点的偏移。
- 性能瓶颈：VDM 的最大问题在于其输出层规模与顶点数量成正比。对于一个拥有 5 万顶点的角色，输出层需要 15 万个神经元。这不仅导致模型文件体积巨大（显存占用高），而且在推理时需要消耗极大的显存带宽（Memory Bandwidth）来写入顶点缓冲区。此外，该模型无法利用现有的硬件蒙皮加速单元，必须完全依赖通用的计算着色器（Compute Shader）[12]。

3.2 Neural Morph Model (NMM)：工业级标准的建立

Neural Morph Model 是目前 UE5 中最为成熟且推荐使用的架构。它通过引入稀疏性（Sparsity）和复用现有的渲染管线特性，解决了 VDM 的瓶颈 [14]。

3.2.1 架构原理：预测权重而非位置

NMM 并不直接预测顶点位置，而是预测一组预计算的“变形目标”（Morph Targets）的权重。

$$P_{final} = P_{LBS} + \sum_{i=1}^{k} w_i \cdot M_i$$

其中 w_i 是神经网络预测的权重，M_i 是在训练阶段通过聚类算法（如 PCA 或 K-Means）提取出的基础变形形态（Basis Shapes）。

- 压缩优势：由于 k（变形目标数量，通常为几百个）远小于顶点数量 N，神经网络的输出层规模被压缩了几个数量级。
- 管线复用：变形目标的混合（Morph Target Blending）是图形API（DirectX/Vulkan）和 GPU 硬件高度优化的操作。NMM 只需在 CPU 或 GPU 上推理出少量的权重值，随后的顶点变形计算可以完全利用现有的蒙皮管线，极大提升了效率 [14]。

3.2.2 局部模式 (Local Mode) 与全局模式 (Global Mode)

NMM 提供了两种训练策略：

- 局部模式：为每个骨骼或骨骼群（Bone Group）训练独立的小型网络。这种模式利用了变形的局部性原理（Locality），即左臂的运动几乎不影响右腿的变形。局部网络参数少，推理快，且能更好地捕捉高频细节 [12]。
- 全局模式：训练一个大型网络处理全身。虽然能捕捉长距离关联（如举手拉动衣角），但往往会因为参数平均化而丢失局部细节。

3.3 Nearest Neighbor Model (NNM)：高频细节的非参数化补偿

对于布料等具有复杂褶皱的材质，简单的线性混合（无论是 LBS 还是 Morph Targets）往往难以表现其高频细节。Nearest Neighbor Model 引入了“记忆”机制 [14]。

3.3.1 数学原理：PCA 空间检索

NNM 的推理过程分为两步：

1. **PCA** 预测：神经网络预测当前姿态对应在 PCA 降维空间中的系数（Coefficients）。
2. 最近邻查找：利用预测的系数，在预先存储的训练数据库中查找最接近的 k 个样本。
3. 细节传递：将检索到的样本的高频残差（High-frequency Residuals）叠加到结果上。公式如下：

$$\Delta_{final} = \mu + (C_{pred} \times B) + \Delta_{neighbor}$$

其中 μ 是均值，C_{pred} 是预测系数，B 是 PCA 基底，$\Delta_{neighbor}$ 是从数据库中检索到的细节修正项[17]。

- 适用场景：这种方法特别适合模拟宽松衣物（Loose Clothing），因为衣物的褶皱状态往往具有多模态性（Multimodality），NNM 通过直接检索真实模拟数据，避免了神经网络回归产生的平滑效应（Smoothing Effect）。

---

4. 降维理论在 ML Groom 中的应用：维度灾难的破解

毛发（Groom）数据的维度远超皮肤网格。一个典型的数字人可能拥有 10 万根引导曲线，每根曲线由 20-50 个控制点组成，总自由度达到百万级别。直接对如此庞大的数据进行 ML 训练会导致“维度灾难”（Curse of Dimensionality）。因此，降维（Dimensionality Reduction）是 ML Groom 流程的核心。

4.1 主成分分析 (PCA) 的线性子空间映射

在 Houdini 的 ML Groom 流程中，PCA 是最常用的降维手段[7]。

4.1.1 奇异值分解 (SVD) 与特征向量

PCA 的核心是对训练数据的协方差矩阵进行奇异值分解。

$$X = U\Sigma V^T$$

由此得到的特征向量（Eigenvectors）构成了毛发变形的“主成分”。前 k 个主成分通常能解释 95% 以上的方差（Variance）。

- 物理意义：第一主成分通常对应整体的重力下垂或惯性摆动；后续主成分则逐步对应局部的分裂、卷曲变化等细节[18]。
- 发明变形目标 **(Invented Blendshapes)**：在 Houdini 中，这些主成分被视为“发明的变形目标”。神经网络不再预测百万个点的坐标，而是预测这 k 个（通常为 256-1024 个）主成分的标量权重[7]。

4.1.2 局限性：线性假设的失效

PCA 假设数据分布在一个线性子空间（Linear Subspace）中。然而，毛发的运动包含大量的旋转和碰撞，这些本质上是非线性的流形（Manifold）。例如，一缕头发绕过肩膀的动作，在 PCA 空间中可能需要大量的线性组合来近似，导致效率降低或产生“幽灵穿插”[18]。

4.2 自动编码器 (Autoencoder) 的非线性流形学习

为了突破 PCA 的线性限制，学术界和高端研发管线开始转向自动编码器 [18]。

4.2.1 潜在空间 (Latent Space) 的非线性映射

自动编码器由编码器（Encoder）和解码器（Decoder）组成。

- 编码器：$z = f(x)$，将高维毛发数据压缩为低维潜在向量 z。
- 解码器：$x' = g(z)$，从潜在向量重建毛发数据。通过在层间引入非线性激活函数（如 ReLU, Tanh, SiLU），自动编码器能够"折叠"空间，学习到数据分布的弯曲流形结构 [18]。

4.2.2 性能与精度的权衡

特性	PCA	自动编码器 (Autoencoder)
数学本质	正交线性变换	非线性函数逼近
训练成本	极低（矩阵分解）	高（梯度下降迭代）
推理开销	极低（矩阵乘法）	较高（多层网络前向传播）
重构质量	易丢失高频旋转细节	能更好保持复杂结构 [18]
UE5 实现	简单（Deformer Graph 矩阵运算）	复杂（需 NNE 调用推理引擎）

在实时应用中，目前的工业标准仍倾向于 PCA，因为它能直接通过 UE5 的 Deformer Graph 中的矩阵运算节点高效实现，无需调用繁重的 NNE 推理后端 [7]。但随着 NNE 的优化，Autoencoder 正逐渐成为处理复杂长发 Groom 的未来方向。

---

5. 实时性能优化与渲染管线集成

将训练好的模型部署到 UE5 并保持 60FPS+ 的帧率，需要对渲染管线的各个环节进行深度优化。

5.1 骨骼遮罩 (Bone Mask) 与稀疏更新策略

在 Neural Morph Model 中，全连接层的计算量虽然已大幅压缩，但在多角色场景下依然可观。UE 5.3 引入的 **Bone Mask** 技术通过利用生物运动的稀疏性来优化推理过程 [12]。

- 原理：并非所有顶点的变形都受所有骨骼影响。Bone Mask 允许开发者显式定义哪些骨骼组（Bone Groups）影响哪些神经网络输出。

- 性能收益：当角色仅移动上半身时，系统可以完全跳过下半身相关神经网络分量的计算。实测数据显示，合理配置 Bone Mask 可带来约 40% 的 GPU 和内存性能提升 [12]。

5.2 光线追踪加速结构 (BLAS) 的动态更新挑战

ML Deformer 对顶点位置的修改发生在 GPU 的计算着色器阶段，这意味着每一帧的几何体都在发生变化。对于光线追踪（Ray Tracing）渲染管线，这构成了巨大的挑战 [24]。

- **BLAS 重建 (Rebuild) vs 更新 (Update)**：光线追踪依赖底层加速结构（BLAS）。对于静态物体，BLAS 只需构建一次。对于变形物体，BLAS 必须每帧更新。
- **Refit 策略**：为了避免每帧完全重建（Full Rebuild）的高昂开销，UE5 通常使用 **Refit** 操作，即仅更新包围盒层级而不改变拓扑结构。然而，如果 ML Deformer 产生的变形过于剧烈（如极端的拉伸或挤压），Refit 可能会导致包围盒退化，从而降低光线遍历（Traversal）效率 [24]。
- 控制变量：通过调整 r.RayTracing.DynamicGeometry.MaxUpdatePrimitivesPerFrame，可以限制每帧更新的图元数量，防止在大量角色同时变形时 GPU 帧时间剧烈波动 [24]。

5.3 神经推理引擎 (NNE) 与计算着色器的异构融合

UE5 的 NNE（Neural Network Engine）提供了一个统一的接口来运行 ONNX 模型，支持 CPU、CUDA 和 DirectML 等多种后端 [22]。

- 异构计算：在理想的架构中，轻量级的权重预测网络运行在 NNE（可能利用 NPU 或 Tensor Cores）上，而大规模的顶点位置修正（Morph Target Accumulation）则通过 Compute Shader 在光栅化单元附近的流处理器上执行。这种异构融合最大限度地减少了数据在不同内存池之间的搬运（Data Movement），是实现高性能变形的关键 [22]。

---

6. 碰撞处理与后处理修复逻辑：弥补 ML 的不足

机器学习模型本质上是一个概率预测器，它无法像物理引擎那样严格保证“互不穿透”的硬约束。因此，在 ML 预测之后，必须引入实时的物理纠偏机制。

6.1 符号距离场 (SDF) 碰撞投影

针对毛发与身体的穿插问题，最有效的实时解决方案是基于 符号距离场（**Signed Distance Field, SDF**）的投影 [30]。

- 离线准备：在 Houdini 中生成角色身体的 SDF 体积纹理。
- 实时纠偏：在 UE5 的 Niagara 模块或 Deformer Graph 中，采样当前毛发顶点在 SDF 中的值。如果值为负（表示在物体内部），则沿着 SDF 梯度的方向（即法线方向）将顶点推到表面。
- 优势：SDF 查询是 $O(1)$ 复杂度的操作，非常适合在 GPU 上对数百万个毛发控制点并行执行 [32]。

6.2 速度平流 (Velocity Advection) 修复

另一种更为柔和的修复方法是 速度平流。在 Houdini 生成训练数据时，除了位置差，还可以计算一个“纠偏速度场” [7]。

- 原理：利用皮肤法线和引导曲线切线的叉积构建一个速度场，该场指示了毛发应该如何“流动”以避开碰撞。
- 推理：在 UE5 运行时，根据预测的权重混合这个速度场，并将其作为微小的偏移量叠加到最终位置上。这种方法比 SDF 投影更平滑，不易产生“抖动”伪影。

6.3 Deformer Graph 的可扩展性

UE5 的 **Deformer Graph** 不仅仅是一个播放器，它是一个可编程的变形管线 [34]。技术美术可以通过编写自定义的 HLSL 内核（Kernel）节点，将 ML 的输出与传统的数学函数（如正弦波风力、程序化噪波）混合。

- 应用实例：可以编写一个 Kernel，检测 ML 预测出的眼皮顶点距离，如果小于阈值（即使 ML 预测它们分开），则强制闭合，从而确保闭眼动画的绝对严谨性 [22]。

---

7. 结论与行业前瞻

从 Houdini 到 UE5 的 ML Deformer 与 ML Groom 流程，标志着实时角色制作技术的一次维度飞跃。它将离线物理模拟的精细度（通过 FEM/Otis 获得）与实时渲染的高效性（通过 NMM/PCA 获得）完美融合，打破了“高保真”与“实时”之间的零和博弈。

核心技术总结

1. 数据为王：Otis Solver 和高斯姿态采样策略确保了训练数据的物理真实性与空间覆盖率，这是模型成功的基石。
2. 残差学习：无论是 Neural Morph Model 还是 Vertex Delta Model，其核心都在于学习 $LBS + \Delta$ 中的 Δ，极大地降低了学习难度。
3. 降维制胜：对于高维 Groom 数据，PCA 提供了目前最佳的性价比，而 Autoencoder 代表了非线性表现的未来上限。
4. 硬件协同：Bone Mask、BLAS Refit 策略以及 NNE 异构计算，是将理论模型转化为 60FPS 产品级应用的关键工程化手段。

未来展望

展望未来，微分物理（**Differentiable Physics**）将是下一个突破口。通过将物理求解器本身嵌入神经网络的训练循环（如 Physics-Informed Neural Networks, PINNs），我们可以期待模型不仅能“模仿”变形，还能“理解”物理守恒定律，从而在无需海量数据的情况下实现更强的泛化能力 [10]。此外，随着端到端（End-to-End）神经渲染技术的发展，未来的变形器可能不再输出顶点位置，而是直接输出神经辐射场（NeRF）或 3D Gaussian Splatting 的参数，彻底重构实时角色的渲染管线。

引用的著作

1. Vellum vs FEM : r/Houdini - Reddit, 访问时间为 二月 1, 2026, https://www.reddit.com/r/Houdini/comments/174rqpu/vellum_vs_fem/
2. Is it possible to build a true FEM-based muscle system in Houdini instead of relying on Vellum? - Reddit, 访问时间为 二月 1, 2026, https://www.reddit.com/r/Houdini/comments/1lptv96/is_it_possible_to_build_a_true_fembased_muscle/
3. Fem vs Vellum : r/Houdini - Reddit, 访问时间为 二月 1, 2026, https://www.reddit.com/r/Houdini/comments/1cd2qcs/fem_vs_vellum/
4. What's new Muscles & Character Effects - SideFX, 访问时间为 二月 1, 2026, https://www.sidefx.com/docs/houdini/news/21/muscles.html
5. Differences between Otis solver and Vellum Solver - SideFX, 访问时间为 二月 1, 2026, https://www.sidefx.com/docs/houdini/muscles/differences.html
6. CFX - Muscle and Soft Tissue with Otis | Houdini 21 | Kai Stavginski - YouTube, 访问时间为 二月 1, 2026, https://www.youtube.com/watch?v=UdtM9yD5pLI
7. ML Groom Deformer - SideFX, 访问时间为 二月 1, 2026, https://www.sidefx.com/tutorials/ml-groom-deformer/
8. Case study: ML Deformer H20.5 - SideFX, 访问时间为 二月 1, 2026, https://www.sidefx.com/docs/houdini/ml/mldeformer.html
9. Implementing a Machine Learning Deformer for CG Crowds: Our Journey - arXiv, 访问时间为 二月 1, 2026, https://arxiv.org/html/2406.09783v1
10. Physics-informed neural networks with residual/gradient-based adaptive sampling methods for solving PDEs with sharp solutions - arXiv, 访问时间为 二月 1, 2026, https://arxiv.org/pdf/2302.08035
11. DeltaPhi: Physical States Residual Learning for Neural Operators in Data-Limited PDE Solving - OpenReview, 访问时间为 二月 1, 2026, https://openreview.net/pdf/b51cb3388f4ec0b16eda42a73a734dbfc28000ef.pdf
12. ML Deformer Framework in Unreal Engine - Epic Games Developers, 访问时间为 二月 1, 2026, https://dev.epicgames.com/documentation/en-us/unreal-engine/ml-deformer-framework-in-unreal-engine
13. unreal.VertexDeltaModel — Unreal Python 5.4 (Experimental) documentation, 访问时间为 二月 1, 2026, http://dev.epicgames.com/documentation/en-us/unreal-engine/python-api/class/VertexDeltaModel?application_version=5.4
14. Experimenting with Daz characters and the ML Deformer - Unreal Engine Forums, 访问时间为 二月 1, 2026, https://forums.unrealengine.com/t/experimenting-with-daz-characters-and-the-ml-deformer/732897
15. Unreal Engine 5.1's Updated ML Deformer Explained - 80 Level, 访问时间为 二月 1, 2026, https://80.lv/articles/unreal-engine-5-1-s-updated-ml-deformer-explained
16. How to Use the Machine Learning Deformer in Unreal Engine - Epic Games Developers, 访问时间为 二月 1, 2026, https://dev.epicgames.com/documentation/en-us/unreal-engine/how-to-use-the-

machine-learning-deformer-in-unreal-engine
17. Nearest Neighbor Model 5.3 | Epic Developer Community, 访问时间为 二月 1, 2026, https://dev.epicgames.com/community/learning/tutorials/2lJy/unreal-engine-nearest-neighbor-model-5-3
18. How Autoencoders Outperform PCA in Dimensionality Reduction - Towards Data Science, 访问时间为 二月 1, 2026, https://towardsdatascience.com/how-autoencoders-outperform-pca-in-dimensionality-reduction-1ae44c68b42f/
19. Principal Component Analysis geometry node - SideFX, 访问时间为 二月 1, 2026, https://www.sidefx.com/docs/houdini/nodes/sop/pca.html
20. How is Autoencoder different from PCA - GeeksforGeeks, 访问时间为 二月 1, 2026, https://www.geeksforgeeks.org/machine-learning/how-is-autoencoder-different-from-pca/
21. Autoencoders vs. PCA: Dimensionality Reduction for Complex Data | by Hassaan Idrees, 访问时间为 二月 1, 2026, https://medium.com/@hassaanidrees7/autoencoders-vs-pca-dimensionality-reduction-for-complex-data-e07d4612b711
22. Deformer Graph roadmap - Rendering - Epic Developer Community Forums, 访问时间为 二月 1, 2026, https://forums.unrealengine.com/t/deformer-graph-roadmap/2616570
23. Unreal Engine 5's ML Deformer Sample Gets Updated - 80 Level, 访问时间为 二月 1, 2026, https://80.lv/articles/unreal-engine-5-s-ml-deformer-sample-gets-updated
24. Ray Tracing Performance Guide in Unreal Engine - Epic Games Developers, 访问时间为 二月 1, 2026, https://dev.epicgames.com/documentation/en-us/unreal-engine/ray-tracing-performance-guide-in-unreal-engine
25. Best Practices: Using NVIDIA RTX Ray Tracing (Updated), 访问时间为 二月 1, 2026，https://developer.nvidia.com/blog/best-practices-using-nvidia-rtx-ray-tracing/
26. microsoft/OnnxRuntime-UnrealEngine: Apply a Style Transfer Neural Network in real time with Unreal Engine 5 leveraging ONNX Runtime. - GitHub, 访问时间为 二月 1, 2026，https://github.com/microsoft/OnnxRuntime-UnrealEngine
27. UNeuralMorphModel | Unreal Engine 5.7 Documentation | Epic Developer Community, 访问时间为 二月 1, 2026, https://dev.epicgames.com/documentation/en-us/unreal-engine/API/Plugins/NeuralMorphModel/UNeuralMorphModel
28. NNE - Neural Post Processing | Epic Developer Community, 访问时间为 二月 1, 2026, https://dev.epicgames.com/community/learning/tutorials/7dr8/unreal-engine-nne-neural-post-processing
29. Curious to know how niche Compute Shaders are. Any of you that knows how to program compute shaders? What about cpu multithread and SoA programming (data oriented programming)? Who here knows what it's like... to Actually be

FAST? - Reddit, 访问时间为 二月 1, 2026, https://www.reddit.com/r/UnrealEngine5/comments/1pees4w/curious_to_know_how_niche_compute_shaders_are_any/
30. Real-time Physically Guided Hair Interpolation - Kui Wu, 访问时间为 二月 1, 2026, https://kuiwuchn.github.io/hairInterp.html
31. Niagara chain physics and collision on permanent particles - Real Time VFX, 访问时间为 二月 1, 2026, https://realtimevfx.com/t/niagara-chain-physics-and-collision-on-permanent-particles/21806
32. Real-to-Sim Deformable Object Manipulation: Optimizing Physics Models with Residual Mappings for Robotic Surgery - arXiv, 访问时间为 二月 1, 2026, https://arxiv.org/html/2309.11656v2
33. Hair Physics in Unreal Engine - Overview - Epic Games Developers, 访问时间为 二月 1, 2026, https://dev.epicgames.com/documentation/en-us/unreal-engine/hair-physics-in-unreal-engine---overview
34. Setting Up a Groom Deformer Graph in Unreal Engine - Epic Games Developers, 访问时间为 二月 1, 2026, https://dev.epicgames.com/documentation/en-us/unreal-engine/setting-up-a-groom-deformer-graph-in-unreal-engine
35. STEP: extraction of underlying physics with robust machine learning - PubMed Central - NIH, 访问时间为 二月 1, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC11296055/