

Study transcript

English narration with Chinese companion explanations (中文讲解).

1. What's New in AI Motion

Good evening, everyone, and welcome to Week 7. Last week Yaofu showed us how quantization makes models run faster on smaller machines. Tonight is different from every other lecture so far. There is no taxonomy and no list of techniques. Instead I will tell you four stories, all of which happened between March and September of this year, and every one of them is about motion: how long a generated video can move, who decides how it moves, whether that motion can leave the screen, and what happens when the tool you used to make it disappears. Each story comes with a demo and a question for you to argue about.

欢迎来到第七讲。上周助教刘耀夫介绍了量化如何让模型更快。今晚不讲分类法、不列技术清单，而是讲四个发生在今年3月到9月之间的故事，都与“运动”有关：生成视频能动多久、谁决定它怎么动、这种运动能否离开屏幕，以及当创作工具消失时会发生什么。每个故事都配有演示和一个讨论问题。

2. Four stories from this year

Here is the map for tonight. Story one: the five-second clip is over. The leading models now generate thirty seconds in one pass, and a research model generates a full minute faster than you can watch it. Story two: music becomes the choreographer. In July, Alibaba released an open model that turns a song and one photo into a minute of dance. Story three: motion leaves the screen. In June, a Korean robotics company taught a humanoid robot a viral K-pop dance using only phone video. Story four: Sora is gone. OpenAI closed the Sora app in April and switched off the API three weeks ago. Between stories two and three we'll have student presentations and a break, and we'll end with a debate.

今晚的四个故事：一、五秒片段的时代结束了——主流模型一次生成30秒，研究模型实时生成一分钟；二、音乐成为编舞——7月阿里开源的模型用一首歌和一张照片生成一分钟舞蹈；三、运动离开屏幕——6月韩国一家机器人公司只用手机视频教会人形机器人跳K-pop；四、Sora消失了——OpenAI 4月关闭应用、9月关闭API。中间有学生展示和休息，最后是辩论。

3. Where we've been: six weeks in one slide

Before the new material, a thirty-second recap, because tonight connects to almost everything we've done. In Week 1 we went from neurons to a full AI filmmaking workflow. Weeks 2 and 3 opened up editing and the generator itself: seeds, guidance and why a model produces what it produces. Week 4 was video and art filmmaking. Week 5 asked who owns a model you adapt. And last week Yaofu showed how quantization makes models faster and cheaper. Tonight's four stories pull on three of those threads at once: how long a video can be, how fast it can be made, and who gets credit for what it shows.

先用一页回顾前六周：第一周从神经网络到AI电影 workflow；第二、三周讲编辑与生成器内部；第四周讲视频与艺术电影；第五周讨论模型归属；第六周讲量化提速。今晚把时长、速度和作者身份这三条线索串起来。

4. How many shots is this?

Let's start with a clip. This is the hero video on ByteDance's official Seedance 2.5 page. Watch it once, and count the cuts. Then tell me: how many times do you think the model was run to make it? Take a guess with your neighbour. [Play the video.] The answer is: the model is built to make up to thirty seconds in a single generation, with camera moves and blocking inside it. A year ago, anything like this was ten or more separate five-second clips stitched together by an editor. Keep that number, thirty seconds, in your head. It is the first story.

先看一段字节Seedance 2.5官方页面上的视频。请数一数镜头切换，并猜猜模型运行了几次。答案是：这个模型可以一次生成最长30秒，内含运镜与调度。一年前，这样的片子需要十几个五秒片段由剪辑师拼接。

5. How top courses treat what's new

Why teach the news? Because the best courses do. Stanford's CS348K this spring, taught by Kayvon Fatahalian, gave a whole lecture to how distillation and caching made autoregressive video models run in real time, and brought in guests from WorldLabs, Roblox and NVIDIA. Jun-Yan Zhu's CMU course put a fast video paper from the same year on its reading list, and ended with forensics and data attribution. MIT's architecture course last fall listed Sora as the video tool in its syllabus, which matters for story four. And Holly Willis at USC pairs every tool with the controversy around it, which is exactly what our 'for the room' questions do tonight.

为什么要讲新闻？因为顶尖课程都这样做。斯坦福CS348K今年春季专门讲了蒸馏与缓存如何让自回归视频模型实时运行；卡内基梅隆16-726把当年的视频论文列入阅读，并以取证和数据归属收尾；MIT建筑系去年秋季的课纲把Sora列为视频工具；南加大的课程把每个工具和它引发的争议放在一起讨论。

6. The 5-second clip is over

Story one. For most of the last two years, an AI video was about five seconds long. That limit shaped everything: how people wrote prompts, how they edited, even what kinds of films they made. This year the limit broke.

第一个故事。过去两年，AI视频大多只有五秒左右，这个限制影响了提示词、剪辑方式甚至影片类型。今年，这个限制被打破了。

7. The unit of AI video is now the scene

Here is the change in four numbers. In 2024, a typical clip was about five seconds. Early this year, ByteDance's Seedance 2.0 got to about fifteen. By this summer, three commercial models, Seedance 2.5, Alibaba's Wan 3.0 and Kuaishou's Kling 4.0, were announced or released at thirty seconds in one pass. And a research model from Peking University and ByteDance, called Helios, makes about a minute, and it makes it faster than real time. Thirty seconds is long enough for a whole scene: an entrance, an exchange, an exit. So the unit of AI video is no longer the shot. It is the scene.

用四个数字看变化：2024年典型片段约5秒；今年初Seedance 2.0约15秒；今年夏天Seedance 2.5、通义万相Wan 3.0和可灵Kling 4.0均支持一次生成30秒；北大与字节的研究模型Helios能生成约一分钟，而且比实时还快。30秒足以容纳一个完整场景，所以AI视频的基本单位从“镜头”变成了“场景”。

8. Who shipped what

Who shipped what. ByteDance launched Seedance 2.5 on July 31st; its official page says it is built for thirty-second storytelling, with camera movement and performance blocking. Alibaba opened a public beta of Wan 3.0 in August, at thirty-second clips, but with no open weights this time. Kuaishou opened early access to Kling 4.0 Flash on September 28th, at twenty seconds and 720p, and has announced the full model for this month. And Helios, from Peking University and ByteDance, came out in March as open weights. One caution: vendors publish few specs, so some of these numbers come from reviewers. Treat them as this month's numbers, not as facts that will last.

各家发布了什么：字节7月31日发布Seedance 2.5，官方称为30秒叙事而生；阿里8月开放Wan 3.0公测，30秒，但未开源；快手9月28日开放Kling 4.0 Flash抢先体验（20秒、720p），完整版计划本月发布；Helios 3月开源。提醒：部分参数来自评测者，只代表本月情况。

9. Why five seconds was the limit

Why was five seconds the wall for so long? Three reasons. First, frames: at twenty-four frames per second, five seconds is a hundred and twenty frames; thirty seconds is seven hundred and twenty, six times as many. Second, attention. Remember from Week 3 that a transformer compares every piece of the video with every other piece. Six times the pieces means roughly thirty-six times the comparisons. That is a rule of thumb, and real systems use many tricks, but it explains why length was expensive. Third, errors pile up: the longer the video, the more chances for a small mistake to grow.

为什么五秒是上限？一、帧数：24fps下5秒120帧，30秒720帧，是6倍；二、注意力：每个token要与所有token比较，6倍的token大约是36倍计算（经验法则）；三、错误会累积。

10. Two ways to make a long video

There are two broad ways to make a long video. One is all at once: the model denoises every frame of the clip together, which keeps the clip consistent but gets expensive fast, and you wait until the whole thing is done. The other is forward in time, called autoregressive: the model writes the next chunk based on the chunks before it, exactly as a language model writes the next word. That lets you stream video and even run in real time, but it has to fight drift, because any error is carried forward. Helios is the second kind.

长视频有两种生成方式：一次性生成所有帧（一致性好，但代价高、要等全部完成）；或沿时间自回归生成（可以流式、实时，但要对抗漂移）。Helios属于后一种。

11. Under the hood: where top courses teach this

Now we go under the hood for about ten minutes. Everything in this technical section is taught at top universities this year, so here is where you can go deeper. MIT's 6.S184 builds a latent diffusion model from scratch with flow matching, and ends with a case study of large video generators. Stanford's CS348K had a lecture this May called 'Performance optimizations for autoregressive video models', covering distillation and KV caching. CMU's 16-726 reading list runs from Video Textures to CausVid, plus the distillation papers. And Stanford's CS231n now opens with generative and interactive visual intelligence. I'll take one idea from each.

接下来十分钟进入技术细节，都是顶尖大学今年在教的内容：MIT 6.S184用流匹配从零搭建潜空间扩散模型；斯坦福CS348K讲自回归视频的性能优化（蒸馏、KV缓存）；CMU 16-726的阅读涵盖CausVid和蒸馏；斯坦福CS231n以生成与交互式视觉智能开篇。

12. Under the hood: from pixels to tokens

First idea, from MIT's course: video models don't see pixels, they see tokens. Take a 720p frame: about nine hundred thousand pixels. A video autoencoder, here Alibaba's Wan-VAE, compresses four times in time and eight by eight in space. Then the transformer cuts that latent into two-by-two patches. Do the arithmetic: 1280 divided by 16 is 80, 720 divided by 16 is 45, so 3,600 tokens per latent frame. Five seconds at 24 frames per second is 120 frames, about 30 latent frames, so roughly a hundred and eight thousand tokens. Thirty seconds is about six hundred and fifty thousand. That number is the whole story of why long video was hard.

第一个概念（MIT课程）：视频模型看到的是token而非像素。720p一帧约90万像素；Wan-VAE在时间上压缩4倍、空间上8×8；再成2×2的patch，每个潜在帧3600个token。5秒约10.8万token，30秒约65万。

13. Under the hood: why length is expensive

Second idea: why length is expensive. Full self-attention compares every token with every other token, so the cost grows with N^2 . For five seconds, a hundred and eight thousand squared is about twelve billion pairs per layer. For thirty seconds, six hundred and fifty thousand squared is about four hundred billion. Six times the length, about thirty-six times the attention work. Real systems cheat this with windowed or sparse attention, caching and compression, and that cheating is exactly what made thirty seconds and real-time minutes possible this year.

第二个概念：为什么长视频昂贵。全自注意力的代价与 N 的平方成正比：5秒约120亿对，30秒约4000亿对，长度6倍，注意力计算约36倍。窗口/稀疏注意力、缓存和压缩正是今年的突破口。

14. Under the hood: diffusion and flow matching

Third idea, in one slide. Diffusion starts from noise and removes a little at each step; the network learns to predict the noise. Flow matching, which MIT's 6.S184 teaches from first principles, learns a velocity instead: at any point between noise and data, which way should I move? If those paths are straight, you need fewer steps. Most of this year's video models use a flow-matching backbone. And the guidance scale you played with in Week 3 works the same way in both.

第三个概念：扩散模型从噪声出发，每步去掉一点噪声；流匹配学习“速度场”，即从噪声到数据该往哪个方向走，路径越直步数越少。今年多数视频模型采用流匹配。

15. Under the hood: the KV cache

Fourth idea: the KV cache, which is why autoregressive video can stream. When the model generates the first chunk, it stores the keys and values from every attention layer. The next chunk attends to that cache instead of recomputing the whole past. If you keep the cache bounded, or compress it as Helios does, the cost of each new chunk stays roughly flat no matter how long the video gets. And frames can leave the model as they're made, so you can watch, or steer, while it generates. It's the same trick that makes chatbots type quickly, and it's the core of Stanford's May lecture.

第四个概念：KV缓存。生成第一段时保存每层注意力的键和值，后续段直接查询缓存而不重算过去；缓存有界或压缩时每段代价基本恒定，画面可边生成边输出。这也是聊天机器人快的原因。

16. Helios: a minute of video, in real time

Helios is the most surprising of these, because it is a research release with open weights. It is a fourteen-billion-parameter autoregressive model, which means it writes the video forward in time, one chunk after another, like a language model writes words. It makes about a minute per generation, and it runs at about nineteen and a half frames per second on a single H100 GPU. Films play at twenty-four, so that is close to watching it be made live. [Play the feature video.] Then open the project page and scroll to the long-video comparisons and the 'verify consistency' section, where a subject leaves the frame and comes back still the same.

Helios最令人惊讶，因为它是开源的研究模型：140亿参数的自回归模型，像语言模型写字一样沿时间一段段往前生成视频。每次约一分钟，在单张H100上约每秒19.5帧，接近边看边生成。请播放官方视频，并在项目页查看长视频对比和“一致性验证”部分：主体离开画面后再回来，依然是同一个。

17. How Helios gets its speed

How is a fourteen-billion-parameter model this fast? Four ideas, all in the paper. First, it remembers less, but smartly: it compresses the history it carries forward, so each new chunk doesn't have to re-read the whole past. Second, the team reports that each step costs about as much as a 1.3-billion-parameter model, ten times smaller. Third, a distilled version uses just three denoising steps instead of dozens, which is last week's theme again: the same model, less work. And fourth, it is lean to train: up to four of these models fit in eighty gigabytes of GPU memory. [Play the inference speed video on the project page.]

Helios为何这么快？一、压缩历史上下文，不必重读整个过去；二、每步代价接近13亿参数模型；三、蒸馏版只需3步去噪（吕应上周的主题）；四、训练高效，80GB显存可放四个14B模型。

18. Under the hood: distillation, from 50 steps to 3

How do you go from fifty steps to three? Distillation. You keep the slow, high-quality model as a teacher, and train a student to jump straight to where the teacher would end up. Two families of methods appear on CMU's reading list: consistency models, and distribution matching distillation, or DMD. Helios's distilled version uses three steps. Connect this to last week: Yaofu's quantization makes each step cheaper; distillation means taking fewer steps. Real-time systems use both.

蒸馏：保留慢而高质量的教师模型，训练学生模型直接跳到教师的终点，步数从几十步降到1-4步（Helios蒸馏版为3步）。上周的量化让每步更便宜，蒸馏让步数更少。

19. Under the hood: a real-time budget

Put it together as a budget. At twenty-four frames per second, you have about forty-two milliseconds per frame. Helios at 19.5 frames per second

is spending about fifty-one. To hit a deadline like that, you have four levers. Fewer steps, through distillation. Cheaper steps, through quantization, which was last week. Less context, through caching, compressed history and sparse attention. And smaller token counts, through stronger compression. Every real-time video system you'll see this year pulls some combination of these four levers.

把它当作预算：24fps每帧约42毫秒，Helios的19.5fps约51毫秒。四个杠杆：更少步数（蒸馏）、更便宜的步骤（量化）、更少上下文（缓存、稀疏注意力）、更少token（更强压缩）。

20. Why long video falls apart: drift

Why was the limit five seconds for so long? Because of drift. When a model writes video forward in time, every chunk is built on the one before it. A small mistake in second one, a slightly wrong shade of a jacket, gets copied and amplified. By second sixty the face, the clothes and the room have quietly changed. Earlier systems added tricks at playback time to fight this. The Helios team did something simpler to explain: during training, they deliberately showed the model drifted video, so it learns to pull itself back. It's like training a tightrope walker by pushing them off balance on purpose. Stanford's CS348K made the same point this spring: real-time video came from training choices and caching, not only from bigger GPUs.

为什么长视频会崩？因为“漂移”。模型沿时间往前生成，每段都基于上一段，第一秒的小错误会被不断放大，到第六十秒脸、衣服和房间都变了。Helios的做法是在训练时故意给模型看“已经漂移”的视频，让它学会自己纠正，就像训练走钢丝时故意推一推。

21. Under the hood: the train-test gap

Here is the technical name for drift: the train-test gap, or exposure bias. With teacher forcing, the model learns each chunk from perfect, real previous frames. But when you use it, it only has its own imperfect frames, so it's always slightly out of its comfort zone, and errors compound. The field has attacked this in steps. Diffusion Forcing in 2024 trained with different noise levels on different frames. In 2025, CausVid distilled a fast causal generator, and Self Forcing, from Adobe and UT Austin, trained the model on its own rolled-out frames using the KV cache. Helios this year simulates drift directly in training. Four papers, one problem.

漂移的技术名称是训练与测试之间的差距（暴露偏差）：训练时用真实的前序帧，使用时只有自己生成的不完美帧。Diffusion Forcing（2024）、CausVid与Self Forcing（2025）、Helios（2026）依次攻克这一问题。

22. Test it yourself: does she come back the same?

Here is the test I want you to apply to any long video: when something leaves the frame and comes back, is it still the same? On the Helios page there's a section called 'Verify Consistency' that does exactly this. [Play one.] Then scroll to the interactive showcases, where the text prompt changes while the video keeps going. That second one is the start of something new: a video you can steer while it plays.

用这个方法检验任何长视频：主体离开画面后再回来，还是同一个吗？Helios项目页的“一致性验证”部分正是做这个；“交互视频”部分则在视频播放中改变提示词。

23. Real time changes what video is

Why does real time matter so much? Because it changes what video is. If frames arrive faster than you can watch them, you're no longer watching a file. You can steer it while it plays, like a game. Stanford's CS348K gave a whole lecture this May on how distillation and caching made this possible. And this is where we're heading in Week 10, on November 3rd, when we look at interactive world models and explorable art. Keep Helios in mind until then.

实时生成改变了视频的本质：如果画面生成得比观看还快，就可以边看边操控，像游戏一样。斯坦福CS348K今年5月专门讲过；第十周（11月3日）我们将讨论交互式世界模型。

24. The long take is an old dream

None of this is new as a dream. Filmmakers have chased the long take for decades. In 1948 Hitchcock made *Rope* from a handful of very long takes and hid the cuts. In 2002 Alexander Sokurov's *Russian Ark* was filmed as a single continuous take of about ninety minutes, through the Hermitage museum. In 2019 Sam Mendes's *1917* was built to look like one continuous shot. Each of those took months of rehearsal and choreography of camera and actors. What changes in 2026 is who can attempt one, and how cheaply you can fail and try again.

长镜头是电影的老梦想：1948年希区柯克《绳索》用少数长镜头并隐藏剪辑点；2002年《俄罗斯方舟》一镜到底约90分钟；2019年《1917》被设计成一个连续镜头。这些都需要数月排练。2026年改变的是谁能尝试、失败成本有多低。

25. What 30 seconds unlocks: blocking

So what does thirty seconds unlock in practice? The vocabulary on ByteDance's official Seedance 2.5 page is revealing: performance blocking and camera movement. Blocking is a theatre word for where actors stand and move through a whole scene. You can't block a three-second gesture; you can block thirty seconds. The page also lists green-screen editing and what it calls white-model control, which means you bring a rough, untextured version of your scene to steer the result. And you can extend a generation twice. Notice that the language has moved from prompting to directing.

30秒带来了什么？Seedance 2.5官方页面用的词很说明问题：表演调度（blocking，演员在整个场景中的走位）、运镜、绿幕编辑和白模控制，还可以续写两次。语言从“提示”变成了“导演”。

26. Useful: a beat sheet for a 30-second generation

Something practical you can use this week. Thirty seconds is long enough that you need structure, so here is a simple beat sheet. Zero to five seconds: establish where we are and who is here, one clear image. Five to fifteen: develop one action and one camera move, and describe them in the order they happen. Fifteen to twenty-five: a turn, something changes. Twenty-five to thirty: land it, hold or exit, so the shot can be cut into a longer film. It's a starting template, not a rule. The key habit is to write your prompt in time order, with one subject and one camera idea.

实用模板：30秒生成的节拍表。0-5秒建立场景；5-15秒一个动作、一个运镜；15-25秒转折；25-30秒收尾，便于剪辑。按时间顺序写提示词，一个主体、一个镜头想法。

27. If one generation is a whole 30-second scene, what is the director's job now?

For the room. If one generation is a whole thirty-second scene, what is the director's job now? Is it writing, blocking, choosing? And what did the editor just lose? Editors used to make meaning from cuts between short clips. Talk to your neighbour for two minutes, then I'll take three answers.

讨论：如果一次生成就是一个完整的30秒场景，导演的工作变成了什么？剪辑师又失去了什么？与邻座讨论两分钟。

28. Music becomes the choreographer

Story two. Choreographers have always started from music. This summer, a model did too.

第二个故事。编舞家一直从音乐出发，今年夏天，一个模型也做到了。

29. AI dance has a short, fast history

AI dance has a short but fast history. In 2019, Everybody Dance Now from Berkeley transferred one dancer's moves onto another person: motion copied, not invented. In 2021, Google's AI Choreographer generated 3D dance motion from music, trained on a large dataset of real dancers called AIST++. In 2023, EDGE from Stanford made that motion editable: you could pin a pose and let the model fill in the rest. And this July, Wan-Dancer went straight from music and a photo to a finished video. Each step moved one stage closer to the final image.

AI舞蹈的历史短而快：2019年伯克利《Everybody Dance Now》把一个人的动作搬到另一个人身上；2021年谷歌AI Choreographer从音乐生成3D舞蹈动作；2023年斯坦福EDGE可编辑舞蹈；2026年Wan-Dancer直接从音乐和照片生成成片。

30. Under the hood: where dance models learn

Where do dance models learn? Google's AIST++ dataset from 2021 is the classic: five point two hours of 3D dance, fourteen hundred sequences, ten genres, filmed from nine cameras. It's small, clean and labelled, which is perfect for skeleton models. Pixel models like Wan-Dancer need far more video, so they build on a large video foundation model trained on web-scale data. Web video is huge and varied but noisy, with no 3D and, often, unclear consent. Keep that trade-off in mind for the choreographer question.

舞蹈模型从哪里学？谷歌AIST++（2021）：5.2小时3D舞蹈、1408段、10种风格、9个机位，小而干净，适合骨骼模型；像素模型需要更多视频，依赖大规模网络数据和视频基础模型，但噪声多、授权不清。

31. Wan-Dancer: song in, dance out

Wan-Dancer was released on July 13th by Tongyi Lab, the Alibaba team behind the Wan video models, as open weights under the Apache license. You give it three things: a music track, one photo of the dancer, and a style word. It gives you a dance video longer than a minute, at 720p and thirty frames per second, that hits the beat. It knows five styles: Chinese classical, K-pop, street, tap and Latin. [Open the project page and play one Chinese classical and one tap example.] Watch the feet in the tap example. The model has to land the taps on the drum hits.

Wan-Dancer由7月13日由阿里通义实验室开源发布（Apache 2.0）。输入三样东西：一首音乐、一张舞者照片、一个风格词；输出一分钟以上、720p/30fps、卡在节拍上的舞蹈视频。支持中国古典舞、K-pop、街舞、踢踏舞和拉丁舞。请播放古典舞和踢踏舞示例，注意脚步是否踩在鼓点上。

32. Skeleton then, pixels now

Here is the real change. Earlier systems like AI Choreographer and EDGE output a skeleton: 3D joint positions over time. That's very useful for games and animation, but you still need a character, a costume, lighting and a renderer before anyone sees a dance. Wan-Dancer outputs the finished image: the look, the light and the camera all come from the model. It's much faster to try ideas, but harder to edit afterwards, because there is no skeleton to adjust. Which one you want depends on your project.

真正的变化：早期系统输出骨骼（3D关节运动），还需要角色、服装、灯光和渲染；Wan-Dancer直接输出成片，试验快，但事后难以修改。

33. One dancer, many looks; one song, many dancers

These are reference images from the project page: one person, seven outfits. Because the look comes from the photo and the movement comes from the music, you can swap either one. Same song, different photo: the same dance, performed by a different dancer. Same photo, different song: the same dancer, a new dance. Scroll down to section six on the project page, where they compare with an earlier model called X-Dancer. Earlier music-to-dance models tended to fall apart after about twenty seconds; this one holds together for the whole track.

这是项目页的参考图：同一个人，七套服装。外观来自照片，动作来自音乐，所以两者可以分别替换：同一首歌换照片，就是同一支舞换个舞者；同一张照片换歌，就是同一个舞者跳新舞。项目页第六部分与早期模型对比：以前的模型大约20秒后就会崩，而它能跳完整首歌。

34. The trick: plan the song, then the steps

Here is the one idea worth remembering from the paper. Wan-Dancer works at two levels. The global level looks at the whole song and plans its shape: the intro, the verse, the build, the chorus, the outro. The local level then renders the actual steps for each passage, beat by beat. That is exactly how a human choreographer works. Nobody choreographs a three-minute piece one eight-count at a time without first knowing where the big moment is. The model learned to plan before it moves, and that is why it stays coherent for a minute.

论文最值得记住的一点：Wan-Dancer在两个层面工作。全局层面看整首歌，规划结构（前奏、主歌、铺垫、副歌、尾声）；局部层面再逐拍生成每段的具体动作。这正是人类编舞的方式：先定结构，再编步子。先规划、后动作，所以能保持一分钟的连贯。

35. Under the hood: two ways to stay coherent for a minute

Compare the two minute-long systems we've seen tonight. Helios goes one chunk at a time, each conditioned on a compressed past. That's great for open-ended, interactive video, but nothing plans the ending; structure just emerges. Wan-Dancer plans then renders: a global model handles the long-range structure and local models fill in the detail. That works when the structure is known in advance, and a song is a structure known in advance. Writers know this choice too: write word by word, or outline first. Music gives Wan-Dancer the outline for free.

对比今晚两个分钟级系统：Helios逐段生成，适合开放、交互式视频，结构自然涌现；Wan-Dancer先规划后渲染，全局模型管长程结构，局部模型管细节，适合结构已知的情况——歌曲就是现成的大纲。

36. What the model has to hear

To dance to music, a model has to hear four things. The beat: where the accents fall, because a step or a tap has to land on them. The tempo: how fast the beats come; the same move reads very differently at ninety and at a hundred and forty beats per minute. Phrases and sections: dancers count in eights, and songs have verses and choruses, so a movement idea needs a beginning and an end. And energy: when the music is quiet and sparse, and when it explodes. Watch the examples again with these four in mind.

模型需要“听懂”四件事：节拍（重音在哪里）、速度（每分钟多少拍）、乐句与段落（八拍、主歌、副歌）、能量（何时安静、何时爆发）。

37. Under the hood: how a model hears music

How does a model actually hear music? Four steps. The raw waveform is tens of thousands of numbers per second. A spectrogram turns it into energy per frequency over time, and onsets and beats become visible as vertical stripes. From that you compute features: simple beat and onset curves, or embeddings from a pretrained music model; EDGE used Jukebox for this. Finally, the video model uses cross-attention: each video frame looks at the music features at the same moment. It's the same conditioning idea as text prompts in Weeks 2 and 3, except the condition is aligned in time.

模型如何“听”音乐：波形→频谱图（节拍与起音可见）→特征（节拍曲线或预训练音乐编码器，EDGE用了Jukebox）→交叉注意力（每帧对齐同一时刻的音乐特征）。与文本提示同理，只是按时间对齐。

38. Under the hood: beats and frames

Some arithmetic that explains the tap result. At a hundred and twenty beats per minute, there's a beat every half second. At thirty frames per second, that's fifteen frames between beats. Tap dancers often hit eighth notes, every seven and a half frames, and each frame is about thirty-three milliseconds. So for a tap step to look right, the foot has to land within a couple of frames of the sound. That's why tap is the hardest of the five styles, and the best test.

算一算：120bpm每0.5秒一拍，30fps下相隔15帧；踢踏舞常踩八分音符，每7.5帧一次；一帧约33毫秒。脚步必须在声音前后两帧内落地，所以踢踏舞是最难的检验。

39. Five styles, five tests

The project page shows five styles, and each one is a different test. Chinese classical dance tests flow: does the movement stay continuous through a phrase? K-pop tests precision: are the hits sharp and exactly on the beat? Street dance tests groove: does the body bounce with the rhythm, or just strike poses? Tap tests sound: do the feet land where you hear the taps? Latin tests whether the movement is driven from the hips and core. [Play one of each.] Then tell me which style you think the model handles worst, and why.

五种风格，五种检验：古典舞看流畅、K-pop看精准卡点、街舞看律动、踢踏舞看脚下的声音、拉丁舞看核心发力。看完后说说模型哪种风格最弱。

40. Machines in choreography are not new

Choreographers have invited machines into the studio for a long time. Merce Cunningham used chance procedures, like tossing coins, to decide the order of movements, so that his own habits wouldn't decide everything. In 1991 he premiered *Trackers*, the first dance he made partly with *LifeForms*, an early computer program for animating figures. And in 2019, Wayne McGregor worked with Google on *Living Archive*, an AI tool trained on twenty-five years of his company's videos that suggests the next phrase. His dancers didn't copy it; they translated it through their own bodies. The question isn't whether machines belong in choreography. It's what role they play.

机器进入编舞并不新鲜：默斯·坤宁汉用掷硬币等机遇方法决定动作顺序；1991年的《Trackers》是他首个部分借助LifeForms电脑软件创作的作品；2019年韦恩·麦克格雷戈与谷歌合作的Living Archive会建议下一个舞句，舞者用身体加以转化。

41. Limits to keep in mind

Four limits to keep in mind. Hardware: the weights are open, but this is not a laptop model; its release materials point to datacenter-class GPUs. One photo: everything the photo doesn't show, like the back of the costume, is invented. Whose moves: as viewers, we cannot check whose dances were in the training videos, which matters for the question we'll ask in a moment. And music has rights: the song driving your dance is someone else's work, so credit it and check the licence before you publish.

四个限制：硬件要求高（数据中心级GPU）；外观只来自一张照片，看不到的部分都是编的；观众无法查证训练视频里是谁的舞；音乐有版权，发布前要署名并确认授权。

42. Pitch a one-take

Your turn. In pairs, five minutes. Pitch a thirty-second, single-generation dance piece. You get three things only: one song, and say why that song; one photo, so who or what is dancing, and it doesn't have to be a person; and one sentence describing what happens in those thirty seconds. Then I'll ask three pairs to read theirs aloud. Keep your pitch, because it could become your Assignment 2.

轮到你们了：两人一组，五分钟，构思一个30秒、一次生成的舞蹈作品。只有三样：一首歌（说明理由）、一张照片（谁或什么在跳，不一定是人）、一句话（这30秒里发生了什么）。之后请三组分享。保留你的构思，它可能成为作业二。

43. Who is the choreographer here?

For the room: who is the choreographer of a Wan-Dancer video? The musician, whose song shaped every move? The model and its builders? The thousands of dancers whose videos taught it what K-pop looks like? Or you, the person who picked the song and the photo? Hands up for each. Remember Lecture 5's question, whose model is it. This is the same question, for the body.

讨论：Wan-Dancer视频的编舞是谁？是音乐人、模型及其开发者、训练数据里成千上万的舞者，还是挑选歌曲和照片的你？这是第五讲“模型属于谁”问题在身体上的延续。

44. Motion leaves the screen

Welcome back from the break. Story three. So far every dance tonight lived on a screen. This one doesn't.

欢迎休息回来。第三个故事：到目前为止的舞蹈都在屏幕上，这一个不是。

45. Robots have danced before

Robots have danced before, so what's new? In the 2000s, Japanese researchers taught the humanoid HRP-2 a traditional folk dance, but they needed motion capture of human dancers in a studio. In December 2020, Boston Dynamics' 'Do You Love Me?' video went viral, but that routine was carefully authored by engineers. What's new this June is the input: an ordinary phone video, the kind millions of people post every day. The studio and the engineer have been replaced by a video file.

机器人以前也跳过舞：2000年代HRP-2通过动作捕捉学会日本民间舞；2020年12月波士顿动力《Do You Love Me?》是工程师手工编排的。今年6月的新意在于输入：普通手机视频。

46. A humanoid learns a K-pop challenge from phone video

In June, the Korean company ROBOTIS posted this video of its open-source humanoid, AI Sapiens, dancing the CORTIS 'REDRED' challenge, a viral K-pop routine. The robot is about 1.3 metres tall, weighs 34 kilograms, and has 23 joints. The remarkable part is not the hardware. It is that the robot learned the dance from ordinary two-dimensional video, the kind anyone films on a phone, with no motion-capture studio and no markers on a dancer's body. [Play the video.]

6月，韩国ROBOTIS公布了开源人形机器人AI Sapiens跳CORTIS《REDRED》挑战的视频。机器人约1.3米高、34公斤、23个关节。关键不在硬件，而在它是从普通二维视频里学会的，不需要动作捕捉棚。

47. Four steps from phone to robot

How does a phone video become a robot's dance? Four steps. First, film a dancer. Second, lift the motion: recover the three-dimensional joint

movement from the flat video, the same pose estimation you've seen behind character animation tools. Third, practise: a copy of the robot learns the dance in simulation, in NVIDIA's Isaac Sim, with reinforcement learning fixing balance and timing, because a robot that copies a human's pose exactly will fall over. Fourth, transfer the learned skill to the real robot. ROBOTIS says its actuators were designed to make that last step, from simulation to reality, smaller.

手机视频如何变成机器人的舞蹈？四步：一、拍下舞者；二、从二维视频中恢复三维关节运动（姿态估计）；三、在仿真中用强化学习练习，修正平衡与节奏，因为完全照抄人的姿态会摔倒；四、把学到的技能迁移到真实机器人上。

48. Under the hood: 3D motion from one camera

Step two in more detail: how do you get 3D motion from a single phone camera? First, find 2D keypoints in each frame: wrists, knees, ankles; that part is mature and fast. Then lift to 3D by fitting a body model, usually SMPL, which has twenty-four joints, so that the 2D points make sense as a 3D body. The hard parts: depth is ambiguous from one camera, fast motion blurs, which is exactly what the Lleida study saw with sudden arm moves, and if ground contact is wrong, the feet slide. Remember foot sliding; it comes back in Week 9's evaluation lecture.

单个手机镜头如何得到3D动作：先检测每帧的2D关键点；再拟合人体模型（如SMPL，24个关节）升维到3D。难点：单目深度歧义、快速动作模糊、地面接触不准导致脚滑。

49. Why not just copy the pose?

Why not just copy the dancer's pose onto the robot, frame by frame? Three reasons. A different body: the robot is 1.3 metres tall with 23 joints, and its limbs don't match a human's proportions. Speed limits: motors can't accelerate like muscles. A 2025 study from Lleida that copied a dancer's arm movements onto a humanoid found the robot could reproduce the moves but couldn't keep up with the rhythm. And balance: humans shift their weight without thinking, and a robot that copies a pose exactly simply falls over. That is why step three, practice in simulation, matters.

为什么不能直接照抄姿态？一、身体不同（1.3米、23个关节）；二、速度限制：2025年一项研究发现机器人能模仿舞者手臂动作，但跟不上节奏；三、平衡：照抄姿态会摔倒。

50. Under the hood: retargeting

Getting from the human skeleton to the robot is called retargeting. First, map the joints: the human body model has twenty-four, the robot has twenty-three degrees of freedom, and they don't line up one-to-one, so some are dropped or merged. Second, scale and limit: rescale limb lengths and clamp joint angles and speeds to what the motors can do. Third, fix the contacts with inverse kinematics, so feet land on the floor and hands reach the right place. The output is a reference motion that the robot can at least try to follow.

从人体骨骼到机器人称为重定向：一、关节映射（人体24关节，机器人23自由度）；二、缩放肢体并限制角度与速度；三、用逆运动学修正手脚接触。

51. Under the hood: learning to imitate

Then the robot learns to follow that reference with reinforcement learning. The policy is a neural network that reads the robot's state and outputs motor commands many times a second. The reward says: match the reference, the joint angles, velocities, hand and foot positions and the body's root. Plus penalties for falling, jerky motion and wasted energy, so it dances and survives. This recipe was set by a 2018 SIGGRAPH paper called DeepMimic, and if you want the reinforcement learning underneath, Berkeley's CS 285 is the standard course.

然后用强化学习让机器人跟随参考动作：策略网络读取状态、输出电机指令；奖励是匹配关节角度、速度、手脚位置；惩罚摔倒和抖动。DeepMimic（SIGGRAPH 2018）奠定了这套方法，伯克利CS 285讲解其中的强化学习。

52. The sim-to-real gap

Step three hides the hardest problem in robotics: the sim-to-real gap. The robot's digital twin practises in NVIDIA's Isaac Sim, where it can fall thousands of times for free. But the simulation is never perfect: friction, motor delays and the way parts flex are all slightly different on a real floor. ROBOTIS says it designed its actuators, the motors in each joint, so that simulated joints behave more like the real ones. And notice: a simulator is a model of the world. In Week 10 we'll ask whether video generators can become that model.

第三步隐藏着机器人学最难的问题：仿真到现实的差距。数字孪生在Isaac Sim里练习，但摩擦、电机延迟等与真实世界不同。ROBOTIS用自家执行器缩小这个差距。仿真器就是一种世界模型。

53. Under the hood: domain randomization

One more technique closes the gap from the software side: domain randomization, popularized around 2017. In every practice run, the simulation changes slightly: different friction, different masses, different motor delays, random pushes. A policy that survives a thousand slightly wrong worlds tends to treat the real world as just one more. There's a nice parallel for dancers: rehearsing on different stages, floors and shoes is domain randomization for humans.

软件层面缩小差距的方法：域随机化。每次仿真都随机改变摩擦、质量、电机延迟和外力，能在一千个略有差异的世界里存活的策略，会把现实世界当作另一个。舞者在不同舞台上排练也是同理。

54. Why dance challenges are perfect training data

Why a K-pop dance challenge? Because dance challenges are almost perfect training data. They're short, a complete routine in fifteen to thirty seconds. They're repeated: thousands of people film the same choreography, so there are many examples of one dance. They're on the beat, so the music marks exactly when each move happens. And they're filmed full body, facing the camera, because they're made to be copied by people. That makes them easy for machines to copy too. Think about what that means for the dancers who made them.

为什么是K-pop挑战赛？因为它们几乎是完美的训练数据：短（15-30秒）、重复（成千上万人拍同一支舞）、卡在节拍上、全身正对镜头。为人模仿而拍，也便于机器模仿。

55. A thought experiment

Now a thought experiment. What if step one wasn't a phone video of a real dancer, but a Wan-Dancer video generated from a song? Song, generated dancer, lifted motion, simulation practice, a real robot dancing a dance that no human ever performed. To be clear, this is speculation. As far as public reports go, nobody has shown the whole chain working. But every piece of it now exists, and two of the pieces came out this summer.

思想实验：如果第一步不是真人舞者的手机视频，而是Wan-Dancer根据一首歌生成的视频呢？歌曲→生成的舞者→提取动作→仿真练习→真实机器人跳一支从未有人跳过的舞。这只是推测，公开报道中尚无人完整实现，但每一环都已存在。

56. Video is becoming training data for bodies

Step back from dance for a moment. Video is becoming training data for bodies. For robots, human videos can teach skills without motion capture: dance today, perhaps household tasks tomorrow. For world models, video generators that can predict what happens next may become the simulators robots practise in, which is Week 10. And for you as artists: a dance video you post may now have two audiences, the people who watch it and the machines that learn from it.

视频正在成为身体的训练数据：对机器人，人类视频无需动捕就能教技能；对世界模型，预测未来的视频生成器可以当仿真器；对艺术家，你发布的舞蹈视频有两类观众：人和机器。

57. If a generated video can teach a real body to move, is that video still “fake” ?

For the room. We usually call generated video 'fake'. But if a generated video can teach a real robot to move, is it still fake? And what is a dance video for, once machines are its audience as well as people? Two minutes with your neighbour.

讨论：我们常说生成视频是“假的”。但如果它能教会真实机器人动起来，它还是假的吗？当机器也成为观众，舞蹈视频的意义是什么？

58. Sora is gone

Story four is shorter, and it's not about a new capability. It's about a loss.

第四个故事比较短，它不是关于新能力，而是关于失去。

59. Eighteen months as a product

Sora became a product in December 2024. Sora 2 arrived on September 30th last year, with a social app. On March 24th this year, OpenAI announced it was shutting Sora down. The app and website closed on April 26th, and on September 24th, three weeks ago, the API was switched off, with no replacement model named. OpenAI's help page told users to export their videos, because after the final window, the data would be permanently deleted. OpenAI didn't give a single official reason; reporting points to the cost of compute and competition for chips. One of the best-known video tools of 2025 lasted about eighteen months as a product.

Sora于2024年12月成为产品；2025年9月30日推出Sora 2及社交应用；2026年3月24日宣布停运；4月26日应用和网站关闭；9月24日API关闭，没有替代模型。官方提醒用户导出作品，否则数据将被永久删除。官方未给出单一原因，报道指向算力成本。

60. Open or closed, this year

Let's sort this year's models by one question: could you still run it in five years? Helios and Wan-Dancer, and the Wan 2.2 family, are open weights: you can download the files and keep them. If you keep a copy, you can run them in five years, as long as you can find the hardware. Seedance 2.5, Wan 3.0 and Kling 4.0 are closed: you use them through an app or an API, so they exist only as long as the company keeps them running. And Sora 2 is already gone. This isn't about which is better; closed models are often stronger. It's about what you depend on.

按一个问题给今年的模型分类：五年后还能运行吗？Helios、Wan-Dancer和Wan 2.2是开源权重，保存副本就能；Seedance 2.5、Wan 3.0和Kling 4.0是闭源，取决于公司；Sora 2已经没了。闭源往往更强，关键是你依赖什么。

61. Under the hood: can you run an open model?

Open weights only help if you can run them. Quick arithmetic: a fourteen-billion-parameter model, like Helios or Wan-Dancer, at sixteen bits is two bytes per parameter, so twenty-eight gigabytes for the weights alone. At eight bits, fourteen. At four bits, seven. Activations, the VAE and the

text encoder need memory on top of that. This is why last week's quantization lecture matters for artists too: it's what makes keeping and running open models practical on hardware you might actually have.

开源权重重要能跑起来才有用：140亿参数在16位下仅权重就要28GB，8位14GB，4位7GB，另外还需激活、VAE和文本编码器的显存。这就是上周量化对艺术家也重要的原因。

62. This has happened before: Flash

This has happened before. On December 31st, 2020, Adobe ended support for Flash Player, and two decades of web animation, games and net art stopped playing in browsers. A lot of it was rescued by emulation: the Internet Archive keeps thousands of Flash works playable using an emulator called Ruffle. And art institutions like Rhizome, whose ArtBase has collected born-digital art since 1999, show that preservation is a practice someone has to choose. Your AI films are the next generation of this problem.

这种事以前发生过：2020年12月31日Adobe停止支持Flash，二十年的网页动画和游戏无法播放；互联网档案馆用模拟器Ruffle保存了大量作品；Rhizome的ArtBase自1999年起收藏数字原生艺术。保存是一种主动的实践。

63. Last fall, a top syllabus was built on Sora

This is not someone else's problem. MIT's architecture course on motion graphics, last fall, listed Sora as the video tool in its syllabus. That was a perfectly reasonable choice at the time. Twelve months later, a project made in that class cannot be re-run, re-edited or extended with the same tool. If you made something with Sora last year, the same is true for you.

这不是别人的问题。MIT建筑系去年秋季的动态图形课程把Sora列为视频工具，当时完全合理；一年后，该课的作品已无法用同一工具重新生成或修改。

64. Keep your sources, not just your outputs

So here is the useful lesson. Keep your sources, not just your outputs. Your prompts, word for word. Your seeds and settings. Your references: images, clips, songs, with credit. Your raw generations, before editing. Your edit project files. And the model's name, version and the date you used it. And notice one more thing: open-weight models like Wan 2.2, Helios and Wan-Dancer can be downloaded and run again years from now. A closed app can be switched off.

经验是：保存源材料，而不只是成品。包括原样提示词、种子和参数、参考图片/视频/歌曲（注明出处）、未剪辑的原始生成、剪辑工程文件，以及模型名称、版本和日期。开源模型多年后仍可运行，闭源应用则可能被关闭。

65. A project folder that survives

Here's a concrete template. One folder per project. A README with the model, the version, the date, your settings and one sentence on what you were trying to do. A prompts folder, every prompt you kept, word for word, with its seed. A refs folder for images, clips and songs, with credits and licences. A raw folder with unedited generations, including the interesting failures. And an edit folder with the editing project and the exported master. It takes five minutes to set up. Use it for Assignment 2.

一个具体模板：每个项目一个文件夹。README写模型、版本、日期、参数和创作意图；prompts保存原样提示词和种子；refs保存参考素材及授权；raw保存未剪辑的生成；edit保存剪辑工程和成片。作业二就用这个结构。

66. Useful: pin your versions

One more habit from software engineering. Pin your versions. Write the exact model ID, not just 'Wan': for example, Wan-AI slash Wan-Dancer-14B. On Hugging Face, record the commit hash of the weights you downloaded, because the files under the same name can change. Keep the licence file next to the weights; Apache 2.0 lets you keep and share them. And write down the settings: seed, steps, guidance, resolution and frame rate. With those, someone can make your piece again in five years.

借鉴软件工程：锁定版本。写下完整模型ID（如Wan-AI/Wan-Dancer-14B），记录Hugging Face权重的提交哈希，把许可证放在权重旁，并记下种子、步数、引导、分辨率和帧率。

67. What does an AI artwork need in order to outlive the tool that made it?

For the room. What does an AI artwork need in order to outlive the tool that made it? Compare it with a film, which survives its camera, or a video game, which often doesn't survive its console. What makes the difference?

讨论：AI艺术作品需要什么才能比创作它的工具活得更久？对比电影（比摄影机活得久）和电子游戏（常常随主机消失）。

68. Four stories, one line each

Let's pull it together before the debate. Story one: the unit of AI video moved from the shot to the scene, and toward real time. Story two: a song and a photo now give you a minute of dance, planned the way a choreographer plans. Story three: phone video can teach a real robot to dance, so video now has machine audiences. Story four: tools can vanish, so keep your sources and prefer what you can still run. Now let's argue.

辩论前总结：一、AI视频的单位从镜头变成场景，并走向实时；二、一首歌和一张照片就能得到一分钟的舞；三、手机视频能教机器人跳舞；四、工具会消失，保存源材料。

69. Debate

Debate time. The motion is: a generated dance that a robot can learn is a real dance. The left side of the room argues for, the right side against, whatever you personally believe. Two speakers per side, one minute each, alternating. Use tonight's stories as evidence. Then everyone votes with a show of hands, and we'll see if anyone changed their mind.

辩论环节。辩题：“机器人能学会的生成舞蹈就是真正的舞蹈。”教室左侧为正方，右侧为反方，每方两位发言人各一分钟，最后全体举手表决。

70. Debate starters

If you're stuck, here are some starters. For the motion: choreography is a structure of movement in time, and the model really does plan one; a real body performs it, with real risk; and dance has always borrowed from tools, from Cunningham's chance procedures to software to video. Against: no one intended this dance; a model averaged thousands of other people's dances; the robot executes, it doesn't express; and without a dancer's choice, it's motion, not dance. Use one example from tonight in your minute.

辩论提示。正方：编舞是动作在时间中的结构，模型确实在规划；真实身体在表演；舞蹈一直借助工具。反方：没有人的意图，只是平均了别人的舞；机器人只是执行；没有舞者的选择就只是动作。

71. Useful: try it this week

Here's what you can actually try this week. Helios has a public demo on Hugging Face; you only need a browser. Wan-Dancer has a demo on ModelScope: bring a song you're allowed to use and one photo. Seedance 2.5 is available through ByteDance's apps and API; use the beat sheet from earlier. And if you're curious about the robot side, ROBOTIS documents the AI Sapiens pipeline on its docs site. Whatever you try, start your project folder before your first generation.

本周可以尝试：Helios在Hugging Face有公开演示；Wan-Dancer在ModelScope有演示，准备一首可用的歌和一张照片；Seedance 2.5通过字节的应用和API使用，配合节拍表；ROBOTIS文档介绍了AI Sapiens流程。开始前先建好项目文件夹。

72. This week

This week: three readings. For technique, the Helios paper; read the abstract and look at the figures. For art and technology, the Wan-Dancer project page; watch three examples and decide which one you would call choreography. For culture, a short article on what happened to Sora. If you want the technical side, there is an optional fourth reading: the Self Forcing paper, which goes with tonight's slides on the KV cache and the train-test gap. Quiz 07 is online, with two technique questions, two art questions and one reflection, due with the others at the end of the semester. Assignment 2 is due on October 31st. And next week, Lecture 8, we add sound: native audio-video, performance and real-time characters.

本周三篇阅读：技术——Helios论文（读摘要、看图）；艺术与技术——Wan-Dancer项目页（看三个例子，判断哪个算编舞）；文化——关于Sora停运的短文。另有一篇选读技术论文Self Forcing，对应KV缓存与训练—测试差距两页。第七次测验已上线，学期末一并提交。作业二10月31日截止。下周第八讲：原生音视频、表演与实时角色。

73. All demo links in one place

This slide collects every demo link from tonight, so you can find them again after class. They are also on the course website, on the Lecture 7 page.

本页汇总了今晚所有演示链接，课后也可在课程网站第七讲页面找到。

74. Music, minutes, and machines

To sum up tonight in one line: motion got longer, found a new choreographer, left the screen, and lost a tool. If you remember only one practical thing, keep your sources. Thank you, and see you next week.

一句话总结：运动变长了、有了新的编舞、离开了屏幕，也失去了一个工具。如果只记住一件事：保存你的源材料。谢谢，下周见。

Demo links

- Story 1: [Seedance 2.5 official page](https://seed.bytedance.com/en/seedance2_5) — https://seed.bytedance.com/en/seedance2_5
- Story 1: [Helios feature video \(YouTube\)](https://www.youtube.com/watch?v=vd_AgHtOUFQ) — https://www.youtube.com/watch?v=vd_AgHtOUFQ
- Story 1: [Helios project page](https://pku-yuangroup.github.io/Helios-Page) — <https://pku-yuangroup.github.io/Helios-Page>
- Story 1: [Self Forcing paper \(NeurIPS 2025\)](https://arxiv.org/abs/2506.08009) — <https://arxiv.org/abs/2506.08009>
- Story 2: [Wan-Dancer project page](https://humanaigc.github.io/wan-dancer-project/) — <https://humanaigc.github.io/wan-dancer-project/>
- Story 2: [Wan-Dancer weights \(Hugging Face\)](https://huggingface.co/Wan-AI/Wan-Dancer-14B) — <https://huggingface.co/Wan-AI/Wan-Dancer-14B>

- Story 2: [Living Archive \(Wayne McGregor\)](https://waynemcgregor.com/productions/living-archive) — <https://waynemcgregor.com/productions/living-archive>
- Story 2: [Wan-Dancer demo \(ModelScope\)](https://modelscope.ai/studios/Wan-AI/Wan-Dancer) — <https://modelscope.ai/studios/Wan-AI/Wan-Dancer>
- Story 3: [ROBOTIS: Learning from Videos \(YouTube\)](https://www.youtube.com/watch?v=NeL2BmgcFJs) — <https://www.youtube.com/watch?v=NeL2BmgcFJs>
- Story 3: [AI Sapiens documentation](https://docs.robotis.com/docs/systems/aisapiens/introduction/) — <https://docs.robotis.com/docs/systems/aisapiens/introduction/>
- Story 3: [Study: robot arms copying a dancer](https://doaj.org/article/b55086ceb9bf4dd3b6b2893ef09cb7db) — <https://doaj.org/article/b55086ceb9bf4dd3b6b2893ef09cb7db>
- Story 4: [OpenAI: Sora discontinuation](https://help.openai.com/en/articles/20001152-what-to-know-about-the-sora-discontinuation) — <https://help.openai.com/en/articles/20001152-what-to-know-about-the-sora-discontinuation>