arXiv:2611.30049v1 [cs.SD] 23 Nov 2026

AIRPiano：基于桌面手势的钢琴乐谱跟随

Jinxuan Zhu 和 Zhiyao Duan

罗切斯特大学，美国罗切斯特
jzhu79@u.rochester.edu，zhiyao.duan@rochester.edu

摘要。 AIRPiano 旨在让在桌面上练习钢琴音乐这一常见做法产生声音。用户以已知乐谱为指导，在平面上做出钢琴式手指动作，而无需复现键盘的绝对位置。乐谱提供音高，手部动作则决定演奏哪些乐谱事件以及何时演奏。我们将这种交互形式化为桌面手部动作与符号乐谱事件之间的在线对齐。为解决桌面动作的歧义性、其与钢琴演奏动作之间的差异，以及对齐的动作-乐谱数据稀缺等问题，我们设计了动作与乐谱表示，构建了数据处理流程，并利用现有的钢琴演奏录音训练了一个带监督式注意力的 Transformer。我们进一步设计了一个流式推理框架，将局部对齐预测与序列跟踪相结合，以实现实时交互。我们构建了两个评估集 VSF-Piano 和 VSF-Tabletop，分别用于评估钢琴演奏和桌面演奏中的对齐效果。在进行事后全局时值调整后，AIRPiano 在 VSF-Tabletop 的配对数据子集上、以 133 毫秒为容差时，实现了 74% 的平均和弦对齐率；相比之下，VSF-Piano 上的该指标为 79%。

关键词： 虚拟钢琴 · 乐谱跟踪 · 手部姿态估计 · 人机交互 · 多模态学习。

1 引言

当无法使用钢琴时，钢琴家仍可能希望练习一首乐曲。在这种情况下，他们可以在阅读新乐谱或回忆熟悉曲目的同时，用手指敲击桌面、大腿，甚至在空中做出敲击动作。一个能够将此类动作转换为声音的系统，可以通过帮助钢琴家评估其近似的手指动作与乐谱之间的对应关系来支持练习。该系统还可以在没有实体钢琴、也无需重现精确琴键位置的情况下实现表现性演奏，从而降低技术要求，尤其是在涉及大幅跳跃的段落中。

许多现有的虚拟钢琴系统通过将指尖敲击映射到虚拟键盘或纸质键盘上的琴键，使用户无需实体钢琴即可进行音乐创作 [1, 20, 37, 38, 40]。然而，虚拟键盘或纸质键盘并不总是随手可得，而完整尺寸的键盘布局需要相当大的表面空间。在没有触觉琴键边界的情况下，

准确地按下对应目标音高的琴键也可能具有挑战性。

乐谱引导系统通过提供来自已知乐谱的音高，同时允许演奏者控制时值和力度等方面，为此提供了一种替代方案 [5, 15, 24]。一些

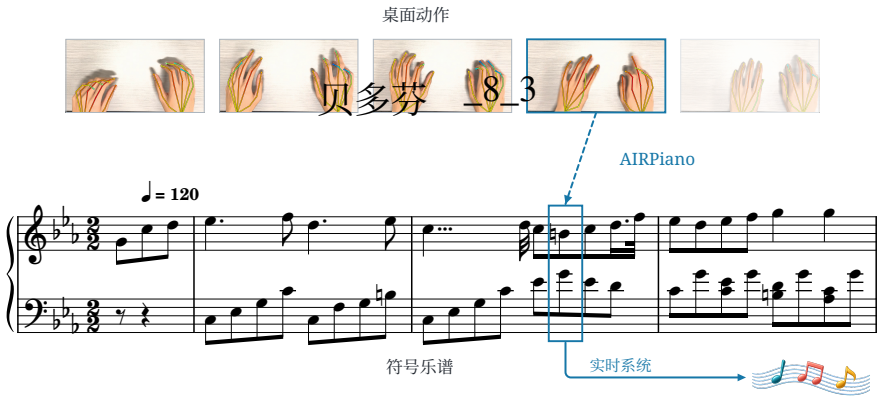

图 1. AIRPiano 系统示意图。乐谱提供音高，而桌面手部动作决定演奏哪些乐谱事件以及何时演奏。虚线箭头将当前视频帧与选定用于实时 MIDI 播放的乐谱事件相连。

系统使用视觉手部追踪和乐谱对齐的排练参考，在实体键盘上跟踪演奏 [27, 34]。

然而，在没有键盘或针对特定曲目的排练参考的情况下，直接将桌面手部动作与符号乐谱事件对齐仍鲜有研究。我们考虑这样一种场景：钢琴家根据已知乐谱排练曲目，在有限的桌面区域内做出钢琴式手指动作，但不追求固定的琴键位置。我们将此任务定义为在线动作-乐谱对齐，这是一种视觉乐谱跟随方法，使用手部动作而非更常见的音频或 MIDI 输入。

这项对齐任务面临若干挑战。主要挑战在于，桌面动作未必保留完整键盘上音高的绝对空间排列。在一个乐句中，敲击位置可能遵循键盘上相对的音高顺序，但同一桌面区域可用于不同的音高范围，因此手部绝对位置并非可靠的对齐线索。此外，手部动作与符号乐谱在表示方式和时间结构上存在显著差异。成对的动作-乐谱数据十分稀缺，而现有钢琴演奏录音中的动作也不同于桌面交互中的动作。解决这些挑战有望在桌面练习过程中提供即时音乐反馈，并为更丰富的表现力控制奠定基础，例如逐音符力度控制。

本文中，我们介绍 AIRPiano，以应对这些挑战。[1] 我们的主要贡献如下：

– 我们将乐谱引导的可听桌面演奏定义为一项在线动作-乐谱对齐任务，并针对这一新任务构建 VSF-Piano 和 VSF-Tabletop 评估集，同时提出事件级评估协议。

[1] 演示可在 https://anonymous.4open.science/w/AIRPiano-90AC/ 查看。代码和完整数据集将在论文被接收后发布。

– 我们构建数据处理流程，从现有钢琴演奏录音中提取对齐的动作与乐谱数据，开发有助于将钢琴演奏迁移至桌面交互的表征，并训练带有监督式注意力的 Transformer，以学习两种模态之间的对应关系。– 我们设计了一种流式推理框架，将局部对齐预测与序列跟踪相结合，以实现实时交互。

2 相关工作

2.1 乐谱引导的视觉钢琴

基于视觉的虚拟钢琴。许多研究探索了适用于不同乐器的虚拟演奏界面。钢琴是其中一个热门例子，计算机视觉的进步使人们能够使用普及的摄像机或红外手部追踪设备捕捉类似弹钢琴的手势。一些系统通过追踪单视角视频中的指尖运动来检测虚拟按键，或使用 RGB-D 摄像机检测无标记平面上的敲击动作 [20, 40]。另一些系统则在纸上绘制键盘，告知演奏者哪些敲击位置对应哪些音高 [1, 37, 38]。

在实体钢琴上，摄像机可以定位单个琴键，并检测琴键何时被按下 [14]。因此，视觉钢琴转录系统可以仅凭视频估计真实钢琴上演奏的音高 [16, 35]。然而，平面上既没有琴键之间的触觉边界，也没有可见的琴键状态变化。这使演奏者更难指定音高，系统也更难识别音高。Matsusue 等人试图将指尖位置与给定的和弦进行相结合，以校正检测到的音高，但可选音高仍仅限于检测到敲击位置对应的琴键或相邻琴键 [22]。Piano Genie 则采用另一种即兴演奏方案，使用学习型解码器将对八个按钮的按压映射为钢琴音符 [8]。这些方法都没有利用完整乐谱来引导演奏。

乐谱引导的演奏界面。 指挥是一门引导合奏团诠释乐谱的艺术。虚拟指挥赋予用户类似的角色，使其能够根据乐谱或录音塑造演奏的速度或力度。*PersonalOrchestra*和*You’re the Conductor* 通过追踪指挥棒的动作来控制管弦乐队录音的回放 [3, 19]。*Home Conducting* 将网络摄像头捕捉到的动作映射为乐谱演奏中的表现力变化 [11]。Mermerci 等人使用长短期记忆网络，根据身体动作估计指挥者在一小节中的相位，然后利用得到的节拍提示调整回放速度 [23]。

让演奏者塑造一首熟悉乐曲的相同理念，也出现在乐谱引导的钢琴界面中。在*Coloring-in Piano*中，每次按键都会触发存储序列中的下一个音高 [24]。iFP 将演奏者的拍时值和力度与演奏模板相结合，由后者塑造每一拍中的音符 [15]。在*Con Espressione!*中，手部动作控制整体速度和响度，同时模型会预测乐谱中每个音符在时值、力度和奏法方面的局部变化 [5]。

IMuSE 的钢琴手势追踪组件与我们的设定更为接近，因为它会追踪钢琴家的手部动作。为了同步伴奏，它将

实时手部动作与依据乐谱 [34] 对齐的预录排练轨迹进行匹配。不过，这要求每首曲目都有一条参考轨迹，而不能仅依靠符号乐谱；此外，钢琴家是在实体键盘上演奏，而不是在桌面上演奏。

2.2 乐谱跟踪

演奏与乐谱对齐将乐谱中的事件与其实际演奏时刻联系起来。离线方法可以利用完整的演奏来匹配单个音符，例如 TheGlueNote，它能够学习 MIDI序列之间的音符相似性 [30]。在线乐谱跟随会跟踪演奏者的演奏过程，从而支持自动伴奏等应用。对于符号输入，相关方法从 Dannenberg 对传入音符的增量匹配 [7] ，到利用离线强化学习训练的跟踪策略 [29]，涵盖了多种方式。基于音频的系统可以根据声学特征在线跟踪乐谱位置 [25]，也可以先转录演奏内容，再将生成的音符事件与乐谱进行对齐 [28]。

与基于音频的乐谱跟踪相比，基于视频的乐谱跟踪受到的关注较少。尽管如此，当音频含糊不清或不存在时，视频仍可提供乐谱位置的依据。与 IMuSE [34], 类似，Pet 和 Maezawa 的系统使用了与乐谱对齐的排练参考。该系统将音频与键盘上的手部和指尖位置相结合，以提高噪声环境中的跟踪效果；同时，他们还仅使用手部关键点进行了评估 [27]。动作还可以为时值提供提前线索。作为未来工作，Raphael 和 Gu 提出利用钢琴琴键运动，在钢琴与管弦乐队同时进入时预测音符击发 [32]。研究还表明，钢琴家在桌面敲击过程中会表现出接触前手指运动 [21]。这些动作可以为交互式系统提供时间，在预期起始时刻之前处理手势。

然而，将桌面手部动作直接与符号化钢琴乐谱进行对齐仍鲜有研究。在桌面上，相同的敲击位置可能对应不同的音高，从而引入空间歧义，而现有基于音频和基于键盘的视觉乐谱跟随方法无法处理这一问题。

3 方法

AIRPiano 使用 Transformer 将观测到的手部动作与乐谱中的音高和节奏模式相结合，并预测局部的动作-乐谱对齐结果。其流式推理框架将这些预测结果与序列跟踪相结合，以跟随钢琴家演奏完整乐谱，并安排和弦播放。该系统根据观测到的运动历史预测即将出现的和弦起始时刻，从而为推理延迟留出时间。

3.1 问题定义

如上文所述，我们将 AIRPiano 的任务表述为在线手部动作—乐谱对齐。我们目前关注起始事件，因为它们标记音符开始的时刻，也是触发乐谱引导回放所需的基本时值事件。令 $S = (s_1、\ldots\ldots、 s_L)$ 表示已知的 L 和弦起始事件序列。乐谱中具有相同量化起始时间位置的音符被归为和弦，事件则按照这些位置排序。

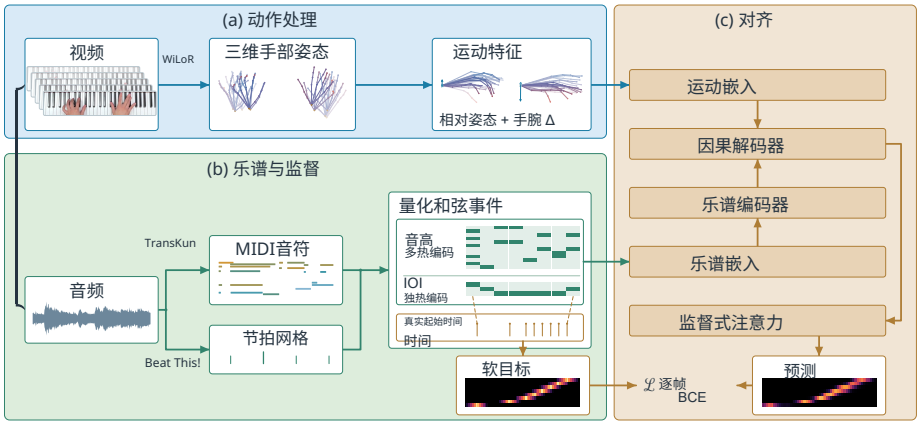

图 2. AIRPiano 的数据处理与训练框架概览。（a）WiLoR 从视频中提取三维手部姿态，并将其转换为相对于手腕的关节位置和手腕位移。（b）TransKun [39] 和*Beat This!* [10] 从对应的音频中提取 MIDI音符和节拍网格。MIDI音符经过量化并分组为和弦起始事件，以音高多热编码和 IOI 独热向量的形式表示，作为模型输入；而原始演奏起始时间则提供软对齐目标。（c）分数编码器和因果运动解码器通过监督式注意力生成帧到和弦的对齐预测。模型使用预测结果与软目标之间的逐帧二元交叉熵进行训练。

令 $m = (m_0, \ldots\ldots, m_{N-1})$ 表示依次接收到的 N 手部动作视频帧序列。我们用包含分配给该帧的事件的集合 A_t 表示帧 t 处的对齐，

$$A_t \subseteq \{s_1, \ldots, s_L\}. \tag{1}$$

在每一帧中，可能触发零个、一个或多个事件。理想的对齐应满足唯一性、覆盖性和单调性：具体而言，$A_u \cap A_v = \varnothing$ 对于 $u \neq v$，$\bigcup_t A_t = \{s_1, \ldots\ldots, s_L\}$，以及 $s_i \in A_u$ 和 $s_j \in A_v$ 与 $i < j$ 意味着 $u \leq v$。然而，模型并未受到显式约束以满足所有这些性质。

在流式处理步骤 t 中，系统使用已观测到的视频历史 $m_{\leq t} = (m_0, \ldots\ldots, m_t)$ 和完整乐谱 S，预测应在 δ 帧后触发的事件，即集合 $A_{t+\delta}$。这里，$\delta \in \mathbb{Z}_{\geq 0}$ 表示提前量（以帧为单位），遵循 [26]。在帧率为 f（目前固定为每秒 30 帧）时，相应的提前量为 $\tau = \delta / f$ 秒。正的 τ 为动作处理、模型推理和声音准备预留了时间，使其能够在预测的起始时刻之前完成。

3.2 动作与乐谱表示

运动。 我们不直接将视频帧 m_t 输入模型，而是首先提取手部姿态，以从输入中去除键盘和背景外观，并为对齐骨干网络提供紧凑的表示。我们使用 WiLoR [31], 这一最新的手部定位与重建模型，按照 MediaPipe [41]的传统，为双手估计 3D 姿态；每只手均由 21 个关节表示。对于

每只检测到的手，WiLoR 都会以物理单位输出完整的关节估计；当手部仅部分可见时，模型会推断不可见关节，只有在完全未检测到手部时，才将其标记为缺失。

令$\mathbf{x}_{t,\,h,\,j} \in \mathbb{R}^3$表示手$h$在帧 t中关节 j的估计位置，以分米为单位，并采用度量相机坐标表示，其中 $h \in \{$左，右$\}$， $j = 0$ 表示手腕。我们还定义二值存在标志 $b_{t,\,h} \in \{0,\,1\}$，其中 $b_{t,\,h} = 1$ 当且仅当手h在帧 t中被检测到时取值为 1，无论其所有关节是否都可见。当 $b_{t,\,h} = 0$时，我们将所有关节的$\mathbf{x}_{t,\,h,\,j} = \mathbf{0}$ 设为零。对于 20 个非腕部关节，我们计算其相对于手腕的位置，

$$\widetilde{\mathbf{x}}_{t,h,j} = \mathbf{x}_{t,h,j} - \mathbf{x}_{t,h,0}, \qquad j = 1, \ldots, 20. \tag{2}$$

由于相似的手指动作可能发生在不同的桌面位置，这种表示能够捕捉局部关节运动，同时减少对手部绝对位置的依赖。

为保留手部平移信息（例如大幅跳跃过程中的移动），我们还计算连续帧之间的腕部位移，

$$\Delta\mathbf{x}_{t,h,0} = \begin{cases} \mathbf{x}_{t,h,0} - \mathbf{x}_{t-1,h,0}, & \text{if } t > 0,\ b_{t,h} = b_{t-1,h} = 1, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3}$$

因此，每只手由关节槽位 0 中的腕部位移以及其余 20 个槽位中的相对腕部位置表示，同时还包含存在标志 $b_{t,\,h}$。

乐谱。 对于每个和弦起始事件 s_i，其音高内容编码为一个 128 维多热向量 $\mathbf{p}_i$，其中各条目表示 MIDI音高 0 到 127 的存在情况。为表示乐谱的节奏结构，我们将每个事件到乐谱时间（即拍）中下一个事件的时间间隔进行编码，使模型能够在根据乐谱对齐手部动作时，同时利用相对时序线索和音高内容。令 q_i 表示事件 s_i 的量化起始刻度，按每拍 24 个刻度计量，以支持常见的二分和三连音细分而无需舍入。其起始间隔（IOI）特征为

$$d_i = q_{i+1} - q_i, \qquad i = 1, \ldots, L - 1. \tag{4}$$

对于最后一个事件，该时间间隔延伸至其音符的最晚结束时刻。我们将 d_i 截断至 $[1, 96]$ 个刻度，并将其编码为一个 96 维独热向量 $\mathbf{r}_i$。

3.3 跨模态对齐学习

模型。 对于每一帧 m_t，我们按照上述方式进行处理，将 $\Delta\mathbf{x}_{t,h,0}$和 $\widetilde{\mathbf{x}}_{t,h,j}$分别对双手进行展平，并附加 $b_{t,h}$，得到一个 128 维向量。MLP嵌入模块将其投影为一个包含双手信息的 512 维标记。对于每个事件 s_i，音高向量 $\mathbf{p}_i$ 和 IOI 向量 $\mathbf{r}_i$ 分别被投影为 512 维嵌入，并将二者相加。每个嵌入模块均由一个线性层、GELU、层归一化和丢弃依次组成。

我们使用Transformer编码器-解码器架构 [36], ，并将正弦位置编码添加到运动标记和乐谱标记中。乐谱标记经过四层

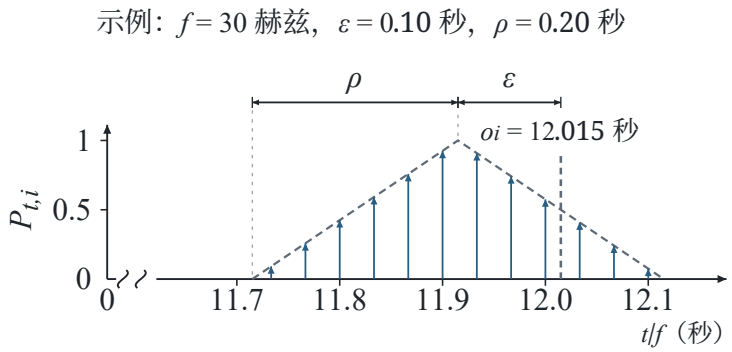

图 3. 三角形起始监督示例。平滑曲线可降低微小时间偏移导致的标签突变，而正值 ε 则鼓励模型利用准备性手部动作进行预期对齐。两者均有助于根据观测到的运动历史 $m_{\le t}$ 预测 $A_{t+\delta}$。

编码器层处理，并采用双向自注意力；运动标记经过四层解码器层处理，在运动帧之间采用因果自注意力，并对编码乐谱采用交叉注意力。因此，每个运动标记都融合了观测到的运动历史和候选乐谱上下文。

受机器翻译 [13] 和语音合成 [33], 中基于注意力的对齐方法启发，我们采用单个监督式注意力评分头进行动作-乐谱对齐。对于包含 T 帧和 C 个候选事件的输入对，令 $\mathbf{u}_t$ 和 $\mathbf{v}_i$分别表示得到的动作与乐谱表示。该评分头使用注意力式缩放点积计算对齐矩阵 $\widehat{\mathbf{P}} \in [0,1]^{T\times C}$，

$$\widehat{P}_{t,i} = \sigma\left(\frac{g_m(\mathbf{u}_t)^\top g_s(\mathbf{v}_i)}{\sqrt{d_a}} + \beta\right), \tag{5}$$

其中，g_m 和 g_s 是学习得到的查询和键投影，维度为 $d_a = 32$；β 是学习得到的标量偏置；σ 是Sigmoid函数。我们采用逐元素 Sigmoid，而非 Softmax归一化，以便每个运动帧上可以有零个、一个或多个事件获得较高分数。我们仍将此输出称为监督注意力矩阵。

监督。 我们根据对齐的演奏构建软目标矩阵 $\mathbf{P} \in [0,1]^{T\times C}$。遵循 Kong 等人 [18],提出的起始回归目标，我们采用三角形时间目标。令 o_i 表示事件 s_i 的参考起始时间（秒）。使用全局运动帧索引 t 和帧率 f，我们定义

$$P_{t,i} = \max\left(0,\ 1 - \frac{|t/f - (o_i - \varepsilon)|}{\rho}\right), \tag{6}$$

其中，$\rho > 0$ 是时间半径，ε 是时间偏移。图 3 展示了由此产生的逐帧目标，以及这一设计背后的动机。

$\widehat{}$我们使用在有效帧-事件 $\widehat{}$对上取平均的二元交叉熵，对**P**进行监督，使其与**P**相匹配，同时排除填充帧和事件。每个预测项 $P_{t,\,i}$ 因此都是一个取值范围为 0 到 1 的起始分数，经过训练后会遵循事件 s_i 在帧 t处的时间目标。在流式推理过程中，这些连续分数会被转换为关于触发哪些乐谱事件的离散决策。

3.4 流式推理

如上所述，模型是在局部乐谱窗口及其对应的动作片段上训练的。为了将其应用于任意时长的流式手部动作，我们借鉴 Dannenberg 的增量匹配和滑动窗口策略，设计了一种流式推理算法 [7]。该算法维护一个乐谱窗口 S_{window}、一段带有起始帧索引 t_{begin}的运动历史 M，以及一个锚点时间映射 A。每收到一帧 m_t，算法便在乐谱窗口和动作窗口上运行模型，根据第 t帧的起始分数触发事件，并按预测的起始时间安排 MIDI 播放。随后，算法使用动态规划跟踪来判定是否将每个触发事件接受为时间锚点，并据此更新 S_{window}。每次更新窗口后，算法都会利用 A 中可用的锚点，更新 t_{begin}，使其与 S_{window} 的起始位置相匹配。该过程持续进行，直至输入的动作帧流结束。下文将详细介绍主要操作。

– **事件触发**。在第 t帧，预测对齐矩阵 $\widehat{}$**P**的最后一行给出了 S_{window}中各事件的起始分数。当某个事件 s_i 的分数超过 0.5 时，该事件便会触发；只有当分数降至 0.5 以下后，才会重新启用触发。这可以防止持续响应导致重复触发。对于公式（6）中定义并在图 3 中展示的三角形轮廓上升段，将目标值 0.5 代入求解，可得交叉时刻为 $o_i-\varepsilon-\rho/2$，即比 o_i提前 $(\varepsilon+\rho/2)$ 秒。因此，理想情况下，可通过选择 $\tau = \delta/f$ 秒来获得名义提前量

$$\varepsilon + \frac{\rho}{2} = \frac{\delta}{f} = \tau. \tag{7}$$

随后，系统会安排每个触发事件的音高 s_i，以$\widehat{o_i} = (t+\delta)/f$ 固定力度和时长进行回放。由于处理开销，第 t 帧的调度决策会在其时间戳 t/f之后作出，但理想情况下应在预定起始时刻 $\widehat{o_i}$之前作出，以留出准备回放的时间。

– **动态规划**。沿用 Dannenberg 针对在线乐谱跟随对最长公共子序列（LCS）问题所作的改编 [7],，我们使用动态规划（DP）逐步将触发事件索引与乐谱事件索引序列（1，2，……，L）进行匹配。具体而言，对于触发事件索引i，令 D_j和 D'_j分别表示更新前后乐谱位置 j处的 DP 分数，初始分数为 $D_j = 0$，边界值为 $D'_0 = 0$。对于 $j = 1,\ \ldots\ldots,\ L$，我们进行如下更新，

$$D'_j = \begin{cases} \max\{D_j,\ D'_{j-1} - 1,\ D_{j-1} + 1\}, & j = i, \\ \max\{D_j,\ D'_{j-1} - 1\}, & j \neq i. \end{cases} \tag{8}$$

匹配一个事件可得1分，跳过一个事件扣1分，而额外的触发事件可以忽略且不受惩罚。我们仅在以下情况下接受触发事件:

DP更新产生新的历史最优值。遵循Dannenberg的窗口更新策略，被接受的事件会提供一个时间锚点A$[i] = \widehat{o}_i$，并将S_{window}重新居中于s_i；否则，该触发事件仍会使窗口向前推进一个乐谱事件。

– **运动历史起始估计**。为与训练设置保持一致，我们需要在每次窗口更新后对齐运动上下文和乐谱窗口的起始位置。更具体地说，我们估计第一个事件的起始时刻 $\widetilde{o}_\ell$ $s_\ell = S_{\text{window}}[1]$。如果可用，我们使用 A $[l]$；否则，根据目标位置 q_ℓ 与锚点的乐谱位置，在相邻锚点之间进行线性插值，或使用最近端点对进行外推。因此，起始帧索引更新为 $t_{\text{begin}} = \lfloor f\widetilde{o}_\ell \rfloor$，并限制在 $[0, t]$范围内。

4 训练

4.1 训练数据

如上文所述，据我们所知，目前没有公开的桌面运动—乐谱数据集。带有符号乐谱的钢琴演奏数据集也十分稀缺，近期的示例包括 SKY-Piano [2]，其中包含 11 小时的录制内容。同步的钢琴视频和音频（例如钢琴演奏的俯视录制视频）则更容易获得。我们使用 PianoMotion10M [12],，该数据集包含从哔哩哔哩收集的 116 小时俯视钢琴演奏视频。我们注意到，这些录制内容主要以流行音乐为主，因此从 PianoYT [17],中补充了古典钢琴演奏；该数据集包含从 YouTube 收集的超过 20 小时的钢琴演奏视频。使用所提供的链接，我们下载视频及其对应的音频。我们选取分辨率为 1920×1080 像素且在可用时帧率为 30 帧每秒的视频；必要时进行重采样，并排除无法访问的录制内容。

我们使用 PianoYT 的训练集划分，以及 PianoMotion10M 的训练集和测试集划分进行训练。由于我们有自己的测试集，因此可以更充分地利用现有录音，尤其是 PianoMotion10M 相当庞大的测试集划分，其规模甚至超过了训练集划分。我们保留 PianoMotion10M 的验证集划分，用于域内评估。

4.2 数据准备

如图 2 所示，动作处理遵循第 3.2 节的流程，训练和推理期间使用相同的处理流程。与此同时，第 3.2 节中的乐谱表示假设存在包含音高内容和量化乐谱位置的和弦起始事件，而在只有同步钢琴视频和音频的数据集中，这些信息无法直接获取。因此，我们尝试通过将转录得到的演奏 MIDI 对照估算的节拍网格进行量化，来转录近似乐谱。我们还保留原始演奏起始时间，以推导第 3.3 节中对齐监督所需的参考起始时间 o_i。

具体而言，我们使用 TransKun 2.0 [39]将每段录音的音频转录为带有起始时刻和结束时刻的 MIDI音符，并使用 *Beat This!* [10] 结合 DBN后处理（忽略强拍标签）来估计节拍时间，如图 2(b) 所示。我们使用每拍 24 个刻度，

将每拍内的候选结束时刻限制为 R ={0、3、4、6、8、9、12、15、16、18、20、21}，以涵盖常见的二分音符和三连音细分。通过在检测到的节拍之间进行线性插值，获得候选时间戳；随后，将每个转录音符的起始时刻和结束时刻映射到时间戳最近的候选刻度。具有相同量化起始时间的音符构成一个事件 s_i，位于刻度 q_i。其参考起始时刻 o_i 是其中各音符原始转录起始时刻的平均值。

4.3 采样与增强

我们以 12 个事件为步长，对包含 $C = 50$ 个和弦事件的乐谱窗口进行重叠采样，并根据参考起始时间提取相应的动作片段。如果某个窗口中检测到任一只手的帧数均未超过 20%，我们便将其舍弃，以过滤视频片头、片尾等无效片段。

对于动作增强，我们分别将每个动作片段的起点和终点随机扰动最多 0.5 秒，以降低对精确窗口边界的依赖。对于乐谱增强，我们将每个片段中的所有 IOI值乘以一个从 {0.5, 1,2} 中均匀采样的共同因子，以降低对其绝对尺度的依赖。

4.4 模型配置 urat和 T r练习设置tup

每个 Transformer 层使用 8 个注意力头，前馈层维度为 2048，并采用 GELU 激活、层前归一化和 0.1 的丢弃率。对于公式（6）中的三角监督，我们设置 $\rho = 0.2$ 秒和 $\varepsilon = 0.1$ 秒，对应于 $\tau = \varepsilon + \rho/2 = 0.2$ 秒的名义提前量。

我们使用 AdamW 训练 20 个周期，峰值学习率为 3×10^{-4}，权重衰减为 0.01，梯度范数裁剪阈值为 1.0。我们在前 5% 的优化器步数中采用线性预热，随后进行余弦衰减。训练使用 BF16 混合精度，有效批次大小为 16，在单个配备 24 GB 显存的 NVIDIA RTX 4090 GPU 上大约需要 8 小时。我们使用最后一个周期的检查点进行评估。

5 评估

5.1 VSF-Piano 和 VSF-Tabletop

桌面演奏不会产生有音高声音，因此难以获取其真实动作—乐谱对齐。为此，我们设计了一套录制与标注流程，让参与者跟随参考钢琴演奏的回放，从而将参考起始时间作为伪真实值。我们采用这套流程构建了两个测试集：VSF-Piano 和 VSF-Tabletop，其中 VSF 是视觉乐谱跟随的缩写。该流程详见图 4。VSF-Piano 包含两位钢琴家的钢琴录音，VSF-Tabletop 则包含四位参与者的桌面录音，其中包括这两位钢琴家。四位参与者都是能够阅读钢琴乐谱的业余钢琴家，所选曲目也都在他们的演奏能力范围内。

我们选择了 10 首涵盖不同音乐风格、速度、节奏模式和音乐织体的曲目。颤音等装饰音在乐谱 MIDI 中被明确实现为音符序列

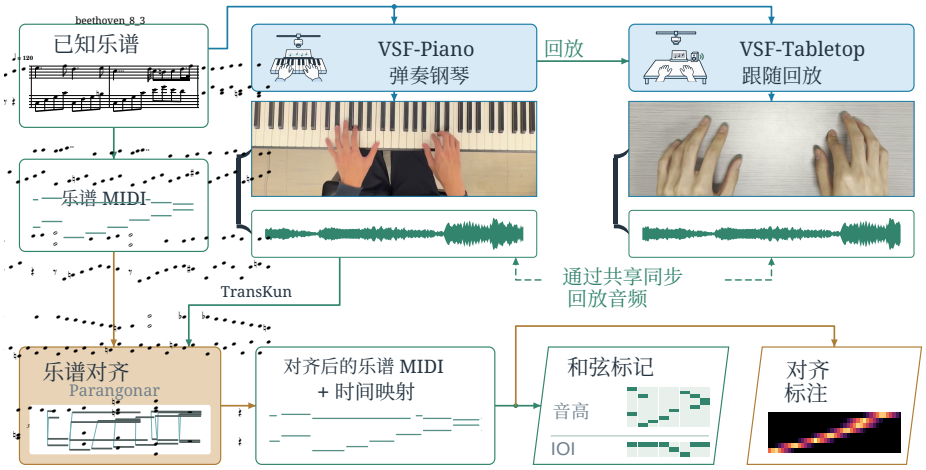

图 4. VSF-Piano 和 VSF-Tabletop 的采集与标注。给定一首已知曲目的乐谱，首先录制一段钢琴演奏；随后，两名桌面参与者（包括演奏者本人）根据同一乐谱进行桌面演奏，尝试通过在桌面上做出自然的钢琴式手指动作来跟上回放的时值，而无需重现键盘上的绝对位置。利用共享的回放音频同步录音，将参考起始时间从钢琴录音传递到桌面录音。接着，我们使用 TransKun [39] 转录钢琴音频，并使用 Parangonar [29] 将乐谱 MIDI 与转录结果对齐，同时允许存在轻微的演奏或转录错误。生成的对齐后的乐谱 MIDI 包含用于对齐标注的、估算演奏时间的乐谱音符，同时通过时间映射保留它们在乐谱中的量化位置，以生成和弦标记。（在实时推理过程中，可以直接从标记了拍点的乐谱 MIDI 获取量化和弦标记；此处的对齐后的乐谱 MIDI 也提供了用于评估的参考起始时间。）

。由于这些知名曲目也可能出现在源训练语料库中，我们使用灵活的动态时间规整（DTW） [4],比较 MIDI序列，同时考虑速度差异和移调，并将识别出的重叠录音从训练数据中排除。录音时长约为 30 至 150 秒，涵盖较短和较长的演奏。所有测试视频均以 1080p、30 帧每秒从俯视角度录制，双手手腕连线大致平行于视频帧的下边缘，与训练数据类似。不同参与者的背景和确切摄像机视角可能有所不同。

5.2 其他评估集

如上所述，我们还使用 PianoMotion10M 留出的验证集划分，对来自与训练数据相同来源的钢琴演奏录音进行域内对齐评估。我们构建了时长分别为 30 秒、60 秒和 120 秒的变体，分别记为 PM30、PM60 和 PM120，以评估模型在不同序列长度下的对齐性能。对于每种时长，我们从每个长度足够的录音中随机提取一个片段。

表 1。VSF-Piano、VSF-Tabletop、VSF-Tabletop（配对）、PM30、PM60 和 PM120 上的宏平均和弦对齐率（C-AR，%），容差分别为 2、4、8 和 16 帧。VSF-Tabletop（配对）包含与 VSF-Piano 中相同两位钢琴家的桌面录音，从而减少参与者构成差异。对于标记为 +5 帧的行，参考起始时间向后移动五帧（在 30 帧每秒下约为 166.67 毫秒），而预测结果保持不变。

评估集	C-AR@2	C-AR@4	C-AR@8	C-AR@16
VSF-Piano	6.10	26.64	80.71	88.90
VSF-Tabletop	9.24	26.71	67.77	76.17
VSF-Tabletop（配对）	6.20	23.20	73.08	83.82
PM30	64.25	88.20	96.06	96.96
PM60	63.48	87.29	96.09	96.92
PM120	65.57	89.01	96.83	97.37
VSF-Piano（+5 帧）	58.83	79.46	85.98	88.79
VSF-Tabletop（+5帧）	44.40	63.94	72.83	76.06
VSF-Tabletop（配对，+5帧）	52.05	74.03	81.26	83.87

模型输入和参考起始时间通过与训练相同的数据处理流程获得。

5.3 评估指标

在我们的桌面动作—乐谱对齐设置下，很难建立一个可直接比较的基线。尽管如此，我们借鉴传统乐谱跟踪中的对齐评估 [6,9]，以提供一种可解释的绝对对齐性能度量。遵循对齐率的定义，我们采用一种和弦级变体，称为和弦对齐率（C-AR），作为主要评估指标，其定义为在指定时间容差内完成对齐的参考和弦事件所占的比例。具体而言，给定通过将起始时间 o_i 量化到帧网格而获得的参考对齐集合 A_t 以及预测对齐集合 $\widehat{A}_t$，令 t_i 和 $\widehat{t}_i$ 分别表示事件 s_i 的参考和预测起始帧，即 $s_i \in A_{t_i}$ 和 $s_i \in A_{\widehat{t}_i}$。对于 k 帧的时间容差，

$$\text{C-AR@}k = \frac{1}{L}\sum_{i=1}^{L} \mathbf{1}\left[\left|\widehat{t}_i - t_i\right| \le k\right], \tag{9}$$

其中，$\mathbf{1}[\cdot]$ 是指示函数，L 是参考和弦起始事件的数量。如果某个事件没有预测对齐，我们令 $\widehat{t}_i = +\infty$，使其计为失败；如果存在多个预测对齐，则采用最早的帧。我们报告各录音的平均 C-AR（宏平均），使每次演奏具有相同的权重。

6 结果

6.1 主要结果

如表 1 所示，我们的模型在域内评估集 PM30、PM60 和 PM120 上的 C-AR@8 均达到 96% 以上。在这些时长下，性能表现相近，

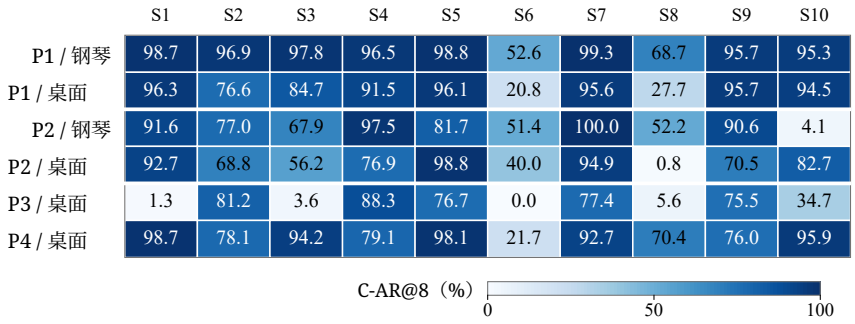

图 5. VSF 数据集上四位参与者（P1 至 P4）和十首作品（S1 至 S10）选段的 C-AR@8（%）热图。P1 和 P2 均贡献了钢琴演奏和桌面演奏，而 P3 和 P4 仅贡献了桌面演奏；后者分别跟随 P1 和 P2 的钢琴演奏回放进行演奏。

表明在所评估的序列长度范围内，流式对齐具有稳定性。在 VSF-Piano 上，C-AR@8 降至 80.71%，说明源录音与我们采集的演奏之间存在领域差距。在 VSF-Tabletop 上，C-AR@8 为 67.77%，而配对数据子集达到 73.08%；与 VSF-Piano 相比，相对降幅仅为 9.45%，证明了模型能够将从钢琴到桌面动作进行零样本迁移。

在 VSF 数据集上，C-AR@2 和 C-AR@4 显著低于 PM 数据集。我们搜索能够使宏平均 C-AR@4 最大化的参考起始时间整数帧偏移。对于 VSF-Piano 和 VSF-Tabletop，将参考起始时间向后移动 +5 帧的偏移均为最优。该偏移显著提高了 C-AR@2 和 C-AR@4，表明存在系统性的时间偏移：预测结果往往比参考起始时间晚约五帧。在 PM60 上未观察到类似偏移，这表明该偏差可能与训练数据和 VSF 数据之间的差异有关。相对于视频，不同录制设备产生的音频延迟差异可能导致这一偏差，因为参考起始时间是根据音频而非视频确定的，而实际动作起始时刻并未进行独立测量。鉴于这些不确定性可能尤其影响严格容差下的评估，我们在主要分析中使用原始、未偏移的参考起始时间，并采用 C-AR@8。

6.2 跨参与者与乐曲的分析

图 5 展示了每个参与者-曲目组合的对齐表现。部分录音接近 100%，表明对齐几乎完整；而另一些则低于 50%，反映出中途失去跟踪等严重失败情况。总体而言，表现似乎更多地因曲目而异，而不是因参与者或钢琴/桌面条件而异。不过，也会出现参与者特有的失败；例如，P3 在 S1（巴赫的 *C*大调前奏曲，BWV 846）中的低分可能与其独特的指法有关，而他在桌面上演奏 S3（贝多芬的 钢琴奏鸣曲第 *8* 号，第三乐章）时省略敲击动作，可能导致了跟踪错误。

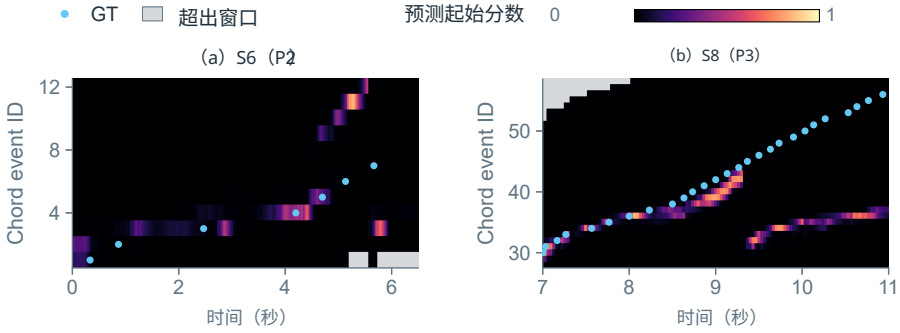

图 6. VSF-Tabletop 上的流式对齐示例。(a) S6 (P2) 从一开始就表现出较弱或错位的响应。(b) S8 (P3) 在快速音阶经过段落中出现跟踪偏差，其中更陡峭的真实值轨迹反映了间隔更近的起始时刻。

表 2. 域内评估集 PM60 以及测试集 VSF-Piano 和 VSF-Tabletop 上各消融变体的 C-AR@8 (%)。

Row	变体	PM60	VSF-Piano	VSF-Tabletop
1	**Ours**	96.09	80.71	**67.77**
2	绝对关节 XY 坐标	95.88	**82.83**	58.42
3	无腕部位移	95.14	70.20	61.49
4	无手部窗口抖动	**96.63**	76.44	60.03
5	无 IOI 特征	92.79	80.29	66.17
6	无 IOI 缩放	94.90	61.48	48.83
7	$\rho = 0$。4 秒，$\varepsilon = 0$	96.55	81.54	66.39

在较具挑战性的曲目中，S6（德彪西的 月光）从一开始就在起始检测方面存在困难（图 6(a)），这表明慢速段落中的细微手部动作提供的视觉起始提示较弱。在 S8（莫扎特的 第 *16* 号钢琴奏鸣曲，第一乐章）中，快速音阶段落会出现跟踪失败（图 6(b)），可能是因为每个音符对应的运动帧数量有限，导致难以区分连续的动作。提高帧率或许有助于捕捉这些快速动作。相比之下，S7（佩措尔德的 *G*大调小步舞曲）和 S4（肖邦的 华丽大圆舞曲，作品 18）总体表现更好，这可能得益于更简单的模式，或稳定的左手伴奏为对齐提供了清晰的提示。

6.3 消融研究

表 2 报告了两个 VSF 数据集上的消融实验结果，并纳入 PM60 以评估域内性能。在运动表示方面，第 2 行将非手腕 XY 坐标替换为 $\mathbf{x}_{t,h,j} - \boldsymbol{\mu}_h$（其中 $j = 1, \ldots, 20$）的 XY 分量，其中 $\boldsymbol{\mu}_h$ 是在训练数据上计算得到的平均手腕位置，用于归一化。保留相对腕部 Z 坐标

是因为绝对深度估计可能不稳定，而腕部位移保持不变。这提高了 VSF-Piano 的性能，但大幅降低了 VSF-Tabletop 的性能，支持了相对手腕坐标在钢琴到桌面的迁移中的作用。在第 3 行中，将 $\Delta\mathbf{x}_{t,h,0} = \mathbf{0}$ 设为会降低整体性能，表明手腕平移能够提供超出局部手指动作之外的有用信息。在第 4 行中，我们禁用了运动片段起止边界各自最多 0.5 秒的独立扰动。这略微提高了 PM60 的性能，但降低了两个 VSF 数据集上的性能，尤其是 VSF-Tabletop，这表明当对齐预测不太可靠时，边界增强可能有助于适应不准确的运动窗口位置。

对于乐谱表示，禁用 IOI 缩放增强（第 6 行）会显著降低 VSF 性能，而完全禁用 IOI 向量的嵌入分支$\mathbf{r}_i$（第 5 行）在 VSF 上的性能相近，但在 PM60 上的性能更低。这一模式表明，模型可能依赖于自动构建的训练乐谱所特有的节奏模式，反映出这些转录乐谱与推理期间使用的真实乐谱之间可能存在差距。IOI 增强或许有助于缩小这一差距。因此，使用与真实乐谱对齐的演奏进行训练，是进一步改进的潜在方向。

最后，将 $\rho = 0.4$ 秒和 $\varepsilon = 0$（第 7 行）设为相应值可获得相近的性能，同时保留 $\tau = \varepsilon + \rho/2 = 0.2$ 秒的名义提前量。不过，允许单独设置目标偏移提供了灵活性：当所需的提前量 τ 发生变化时，我们可以联合调整 ε 和 ρ 以使其匹配，同时控制监督信号的时间分布。

6.4 运行时间与延迟

我们在单个 RTX 4090 上测量每帧处理时间：对于 WiLoR 和 AIRPiano 推理，均使用预热 60 帧后的 1,800 帧。WiLoR 的平均处理时间和第 95 百分位（P95）处理时间分别为 60.45 毫秒和 62.13 毫秒，而 AIRPiano 的对应数值为 6.68 和 6.88 毫秒；AIRPiano 可直接接收提取出的手部姿态。因此，手部提取占据了大部分处理开销。

在 30 帧每秒的帧率下，理想并行执行时，将交替帧分配到两个 GPU 上，可使每个 GPU 处理一帧的时间为 $2 \times 1000/30 \approx 66.7$ 毫秒，超过 WiLoR 的 P95 提取时间。分别测得的 WiLoR 和 AIRPiano 的 P95 处理时间之和小于 100 毫秒。考虑到 $\tau = 200$ 毫秒的提前量，这应当为摄像头采集、音频输出及其他开销留出名义上超过 100 毫秒的预算。我们已经实现了一个实时系统，并使用 WiLoR 的快速推理模式录制了演示，以获得更高的吞吐量和更大的延迟余量。全面的端到端延迟测量和用户研究留待未来开展。

7 结论

我们介绍了 AIRPiano，这是一种由乐谱引导的系统，通过在线动作-乐谱对齐将桌面钢琴手势转换为声音。AIRPiano 结合动作与乐谱表示、监督式对齐学习以及流式序列跟踪，从现有的钢琴演奏录音中学习，以支持桌面交互。在 VSF-Piano 和 VSF-Tabletop 上进行的评估，初步考察了该方法在钢琴演奏和桌面演奏中的表现。

我们的评估重点关注对齐准确度和处理时间，但这些指标可能无法充分反映用户在实时反馈下进行演奏的体验。因此，未来的工作将进一步完善实时系统设计，并开展用户研究，以评估感知延迟以及系统与用户现有演奏习惯的兼容性。我们还将探索表现力控制，例如音符力度、音符关闭时机、装饰音，以及和弦中的起音不同步。

致谢

我们感谢各位志愿者提供钢琴和桌面录音，也感谢协助采购设备的人员。此外，我们感谢 Huiran Yu 推荐相关文献，并感谢 Jiawei Yu 和 Haina Zhu 对稿件撰写提出的宝贵建议。

参考文献

1. Avola, D., Cinque, L., Marini, M.R., Princic, A., Venanzi, V.：Keyrtual：一种基于RGB-D与传感器融合的轻量级虚拟音乐键盘。载于：国际图像与模式计算分析会议。第182–191页。Springer（2023年） 2. Bae, J., Park, D., Kwon, T., Choi, Y.S., Hur, H., Obata, S., Kai, S., Wada, Y., Takahashi, Y., Maezawa, A.等：SKY-Piano：多模态钢琴演奏数据集。arXiv预印本 arXiv:2607.27296（2026年） 3. Borchers, J., Lee, E., Samminger, W., Mühlhäuser, M.：Personal Orchestra：用于交互式指挥的实时音频/视频系统。Multimedia Systems **9**（5），458–465（2004年） 4. Bükey, I., Zhang, J., Tsai, T.：FlexDTW：具有灵活边界条件的动态时间规整。载于：第24届国际音乐信息检索学会会议论文集。第733–740页。ISMIR（2023年11月）。https://doi.org/10.5281/zenodo.10265393，https://doi.org/10 10265393 5 5281。/ zenodo。 5. Cancino-Chacon, C.E., Balke, S., Henkel, F., Stussak, C., Widmer, G.：Con espressione!展览：探索表现性演奏中的人机协作。载于：国际音乐信息检索会议（ISMIR）晚间突破与演示环节。荷兰代尔夫特（2019年） 6. Cont, A., Schwarz, D., Schnell, N., Raphael, C.：实时音频乐谱对齐的评估。载于：国际音乐信息检索学会会议（2007年），https://api.semanticscholar.org/CorpusID:9901746 7. Dannenberg, R.B.：用于实时伴奏的在线算法。载于：国际计算机音乐会议（1984 9472796 8 24 ），https://api.semanticscholar.org/CorpusID: 。 8. Donahue, C., Simon, I., Dieleman, S.：Piano Genie。载于：第届智能用户界面国际会议论文集。第160–164页（2019年） 9. Duan, Z., Pardo, B.：用于在线复调音频乐谱对齐的状态空间模型。载于：2011年IEEE声学、语音与信号处理国际会议（ICASSP）。第197–200页。IEEE（2011年） 10. Foscarin, F., Schlüter, J., Widmer, G.：Beat This! 无需DBN后处理的精确节拍跟踪。arXiv预印本 arXiv:2407.21658（2024年） 11. Friberg, A.：Home Conducting：使用手势控制整体音乐表现力。载于：2005年国际计算机音乐会议论文集（2005年） 12. Gan, Q., Wang, S., Wu, S., Zhu, J.：PianoMotion10M：钢琴演奏中的手部动作生成数据集与基准。载于：国际学习表征会议。第2025卷，第52082–52099页（2025年）

13. Garg, S., Peitz, S., Nallasamy, U., Paulik, M.：使用Transformer模型联合学习对齐与翻译。载于：2019年自然语言处理经验方法会议与第九届自然语言处理国际联合会议（EMNLP-IJCNLP）论文集。第4453–4462页（2019年） 14. Kang, S., Kim, J., Yoon, S.e.：使用计算机视觉的虚拟钢琴。arXiv预印本 arXiv:1910.12539（2019年） 15. Katayose, H., Okudaira, K.：在音乐指挥界面中使用表现力演奏模板。载于：2004年音乐表达新界面会议论文集。第124–129页（2004年） 16. Kim, Y., Sohn, H., Nam, J., Lerch, A.：多任务多帧视觉钢琴转录。arXiv预印本 arXiv:2608.03419（2026年） 17. Koepke, A.S., Wiles, O., Moses, Y., Zisserman, A.：Sight to sound：视觉钢琴转录的端到端方法。载于：ICASSP 2020—2020年IEEE声学、语音与信号处理国际会议（ICASSP）。第1838–1842页。IEEE（2020年） 18. Kong, Q., Li, B., Song, X., Wan, Y., Wang, Y.：通过回归起始时刻与结束时刻实现带踏板的高分辨率钢琴转录。IEEE/ACM《音频、语音与语言处理汇刊》**29**，3707–3717（2021年） 19. Lee, E., Nakra, T.M., Borchers, J.：You’re the Conductor：面向儿童的逼真交互式指挥系统。载于：音乐表达新界面国际会议论文集。第68–73页（2004年） 20. Liang, H., Wang, J., Sun, Q., Liu, Y.J., Yuan, J., Luo, J., He, Y.：徒手音乐：用于虚拟钢琴的实时手部交互。载于：第20届ACM SIGGRAPH交互式三维图形与游戏研讨会论文集。第87–94页（2016年） 21. Loehr, J.D., Palmer, C.：钢琴家的手指敲击中的认知与生物力学影响。《实验脑研究》**178**（4），518–528（2007年） 22. Matsusue, C., Funahashi, K., Mizuno, S.：基于音乐理论校正指法位置偏差的触摸输入式VR钢琴。ACM SIGGRAPH 2017海报（2017年），https: //api.semanticscholar.org/CorpusID:1382968 23. Mermerci, M., Pascoe, E., Edström, F., Kjellström, H.：通过识别指挥手势实时控制虚拟管弦乐团。arXiv预印本 arXiv:2604.27957（2026年） 24. Nishimoto, K., Oshima, C., Miyagawa, Y., Shirosaki, T.：促进音乐表现力的乐器。载于：CHI’02计算机系统中的人为因素扩展摘要。第722–723页（2002年） 25. Park, J., Cancino-Chacón, C., Chiruthapudi, S., Nam, J.：Matchmaker：用于实时钢琴乐谱跟踪和系统性评估的开源库。arXiv预印本 arXiv:2510.10087（2025年） 26. Pasini, M., Nistal, J., Hayes, B., Bjare, M.R., Lattner, S., Fazekas, G.：Liveband：音频领域中的实时伴奏生成。arXiv预印本 arXiv:2606.03803（2026年） 27. Pet, K., Maezawa, A.：将手部姿态估计与音频相结合，实现抗噪钢琴乐谱跟踪。载于：声音与音乐计算会议论文集（2024年），https://kaitlinpet.com/papers/Yamaha_Research.pdf 28. Peter, S., Hu, P., Widmer, G.：将实时钢琴转录与符号级跟踪相结合，实现精确且稳健的乐谱跟踪。arXiv预印本 arXiv:2505.05078（2025年） 29. Peter, S.D.：结合离线强化学习的在线符号音乐对齐。arXiv预印本 arXiv:2401.00466（2023年） 30. Peter, S.D., Widmer, G.：TheGlueNote：用于稳健灵活音符对齐的学习得到的表征。arXiv预印本 arXiv:2408.04309（2024年） 31. Potamias, R.A., Zhang, J., Deng, J., Zafeiriou, S.：WiLoR：野外环境下端到端三维手部定位与重建。载于：2025年IEEE/CVF计算机视觉与模式识别会议（CVPR）。第12242–12254页。IEEE（2025年）

32. Raphael, C., Gu, Y.：自动演奏钢琴的管弦乐伴奏。载于：ICMC（2009） 33. Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., Liu, T.Y.：FastSpeech：快速、稳健且可控的文本转语音。神经信息处理系统进展**32**（2019） 34. Ritter, M., Hamel, K., Pritchard, B.：集成式多模态乐谱跟踪环境。载于：ICMC（2013） 35. Su, K., Liu, X., Shlizerman, E.：Audeo：为无声演奏视频生成音频。神经信息处理系统进展**33**，3325–3337（2020） 36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.：注意力就是你所需要的一切。载于：Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R.（编）：神经信息处理系统进展。第30卷。Curran Associates, Inc.（2017）, https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf 37. Venanzi, V., Princic, A., Marini, M.R., Foresti, G.L., Cinque, L.：Musikeyrtual：在增强现实中演奏音乐键盘的框架。《国际人机交互期刊》**42**（11），7893–7905（2026） 38. Vishal, B., Lawrence, K.D.：纸上钢琴——基于阴影分析的触控交互。载于：2017 2 1 6 nd 国际人机交互会议（MAMI）。页码 –。IEEE（2017） 39. Yan, Y., Duan, Z.：使用非层次化 Transformer 对自动钢琴转录的时间间隔进行评分。arXiv 预印本 arXiv:2404.09466（2024） 40. Yeh, C.H., Tseng, W.Y., Bai, J.C., Yeh, R.N., Wang, S.C., Sung, P.Y.：基于单视角视频和多指动作识别的虚拟钢琴设计。载于：2010 年第三届国际以人为中心的计算会议。页码 1–5。IEEE（2010） 41. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L., Grundmann, M.：MediaPipe Hands：设备端实时手部跟踪。arXiv 预印本 arXiv:2006.10214（2020）