分层子动作树用于连续手语识别

Dejie Yang[1] Zhu Xu[1] Xinjie Gao[1] Yang Liu[1,2]✉
[1]Wangxuan Institute of Computer Technology, Peking University, Beijing, China
[2]State Key Laboratory of General Artificial Intelligence, Peking Universitys, Beijing, China
{ydj,xuzhu,gxj1914779162}@stu.pku.edu.cn, yangliu@pku.edu.cn

arxiv:2506.20947v1 中译本

摘要—连续手语识别（CSLR）旨在将未修剪的视频转录成词汇，这些词汇通常是文本单词。最近的研究表明，由于训练数据不足，缺乏大型数据集和精确标注已成为 CSLR 的一个瓶颈。为了解决这个问题，一些工作开发了跨模态解决方案来对齐视觉和文本模式。然而，它们通常从词汇中提取文本特征而没有充分利用其知识。在本文中，我们提出了分层子动作树（HST），称为 HST-CSLR，以有效地结合词汇知识与视觉表示学习。通过整合来自大型语言模型的特定于词汇的知识，我们的方法更有效地利用了文本信息。具体来说，我们构建了一个 HST 来表示文本信息，并逐步对齐视觉和文本模式，同时受益于树结构以减少计算复杂度。此外，我们施加对比对齐增强以弥合两种模态之间的差距。在四个数据集（PHOENIX-2014、PHOENIX-2014T、CSL-Daily 和 Sign Language Gesture）上的实验验证了我们的 HST-CSLR 的有效性。代码和模型可在以下位置获取：https://github.com/Federfallt/HST-CSLR.git。

***Index Terms*—连续手语识别，层次子动作树，跨模态对齐，大型语言模型**

I. 介绍

手语是聋哑社区的重要交流工具。然而，对于听力正常的人来说，学习和掌握手语可能既具有挑战性又耗时，这在两个群体之间产生了直接沟通的障碍。为了解决这个问题，连续手语识别（CSLR）[1]–[5] 旨在将连续帧转化为一系列词汇单元 [1] 以表达某些句子。

当前的 CSLR 模型 [2]–[5] 通常包括一个空间模块（例如，2D-CNN）以从输入片段中提取短期空间特征，一个时间模块（例如，1D-CNN 或 BiLSTM）来捕捉长期依赖关系，并使用广泛使用的连接时序分类（CTC）损失 [1] 来预测目标手势序列的概率分布。通常，当前的 CSLR 模型可以分为两种方法：多流 CSLR 框架 [6]–[8] 通过扩展空间模块以包含多个线索信息（如关键点热图 [6], [7] 或骨架信息 [8]），来引导模型更加关注手和

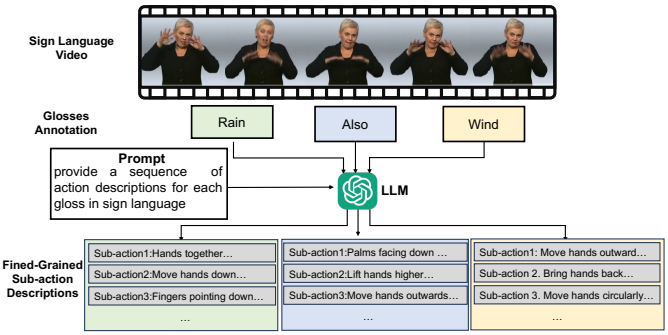

图 1. 手语视频展示了语义信息的层级，从高级事件（注释）到细粒度子动作。然而，现有的数据集缺乏对这些子动作的详细标注，这给提高对手语内容分层理解带来了挑战。为了解决这一问题，我们利用大型语言模型 (LLMs) 生成精确且有意义的子动作描述，从而增强对手语视频的层级理解。

头部的信息，从而获取最重要的语义信息。近期的方法通常侧重于手语视频中的视觉特征，并利用跨模态对齐模块来对齐词汇注释和视觉特征，以提高词汇注释的理解。相比多流框架的复杂性，跨模态对齐框架简单且有效。但是如何更好地利用手势中包含的语言知识仍然是关键问题。尽管取得了巨大的成功，当前的方法并未明确地建模手语视频中存在的复杂的时序和上下文依赖关系，限制了它们的能力，并且如何更好地利用手势中包含的语言知识仍然是关键问题。如图 1 所示，手语视频呈现出层次化的语义结构，范围从高层事件到细粒度子动作，而两个重要的原因阻碍了这种层次化信息的探索：（1）现有的数据集只涵盖了手势级别的注释，并不支持对细粒度行动级别理解的支持；（2）手语中的动作具有明确的时间顺序，其建模被之前的方法所忽视，导致对手语的理解不足。

为此，我们提出了一种名为分层子动作树的连续手语识别新方法（HST-CSLR）。关键思想是通过时间和上下文的角度来改进对手语的理解。具体来说，由于数据集中缺乏细粒度注释，我们建议利用大型语言模型

✉Corresponding author.

[1]手势是手语中的原子词汇单位，通常是单词或短语。

（LLM）探索每个术语表示的潜在信息，获取每个术语的详细和顺序描述。然后为了充分利用这些信息并实现手语的时间建模，我们引入了使用获得的描述构建的分层描述树，并设计了一种树搜索算法，该算法不仅整合了描述知识，还考虑了每个术语的时间信息，从而充分揭示存储在这些描述中的综合信息以跨模式方式辅助对手语的理解。此外，为了明确提高视觉和文本的一致性，我们在激活树节点监督下施加了一种分层跨模态对齐增强方法。我们评估了我们的 HST-CSLR 在四个 CSLR 基准上，包括 PHOENIX-2014 [9]，PHOENIX-2014T [10]，CSL-Daily [11] 和 Sign Language Gesture [12]。定量结果表明我们超越了现有的最先进的工作。通过消融研究，我们也验证了引入模块的有效性。我们的主要贡献如下：(1) 我们开发了一个分层描述树搜索框架，这使得对 CSLR 任务进行全面的上下文和时间建模成为可能；(2) 我们引入了新颖的对齐范式来增强手语视觉与语言模式之间的对齐，从而提高了更稳健的识别性能。我们也补偿了现有手语数据集中细微的术语描述，促进了对手语连续性的进一步理解。(3) 在四个数据集上的实验展示了我们方法的有效性。

II. 相关工作

连续手势语言识别。连续手语识别（CSLR）旨在使用弱监督方法将图像帧序列翻译成相应的词汇表，其中仅提供句子级别的注释作为监督。CSLR 方法可以广泛分为两类：多线索和单线索方法。多线索方法 [7], [13] 结合了多种模式特征，包括手形、面部表情、嘴部动作和姿势。这些方法需要从多个模态中提取并整合特征，导致计算成本高，并且在实际场景中难以应用。为了解决这个问题，最近的研究集中在单线索方法 [4], [14] 上，该方法将单一线索——具体来说是 RGB 帧——作为输入，直接从视频序列解码词汇表。VAC [4] 将视觉编码器视为学生，文本词汇表解码器视为老师，利用知识蒸馏来对齐视觉和文本模态。类似地，CVT-SLR [14] 探索了预训练的视觉和文本模态的知识，使用预训练变分自编码器。在本文中，我们专注于单线索设置以实现高效的训练和推理。

然而，现有的方法未能捕捉到手语视频中固有的时间和语义依赖层次结构，限制了连续手语识别的性能。为了探索手语视频的层次结构和细粒度子动作描述，我们提出了一种分层描述树构建过程，该过程在整合描述知识的同时考虑每个词汇的时间信息。此外，为了增强手语视频与词汇之间的对齐，我们引入了一种对比对齐增强方法。该方法明确提高了视觉和文本的一致性，并为层次树中的节点提供了监督。

III. 提出的方法

A. 问题定义

连续手语识别（CSLR）专注于将图像帧序列翻译成相应的词汇串。该任务通常在一个弱监督环境下进行，其中仅提供句子级别的标注作为监督，而不是逐帧或单词级的标签。给定一个包含 T 帧 $X = \{x_t\}_{t=1}^{T} \in \mathcal{R}^{T\times 3\times H\times W}$ 的手语视频，CSLR 模型旨在将其翻译成词汇串 $Y = \{y_i\}_{i=1}^{N}$，其中 N 表示词汇串的长度并与视频相关。

B. 我们的方法概述

如 Figure 2 所示，我们提出的分层子动作树（HST-CSLR）框架包含四个主要组件：一种基线方法、一个细粒度子动作生成器、一个分层子动作树构建器和一个层次跨模态对齐模块。

1) 基线方法: 基线模型由空间模块、时间模块和分类器组成。具体而言，为了提取手势视频帧的空间-时间表示，空间模块将输入帧处理为帧级别的空间特征，然后时间模块进行时序建模以获得视觉特征 $V \in \mathbb{R}^{T\times d}$。为了从视觉特征计算概率序列，分类器预测每个帧的初始词汇对数几率。基线的目标函数包含三个损失项，CTC 损失 $\mathcal{L}_{seq}$ 用于对数监督，辅助损失 $\mathcal{L}_{VE}$ 以增强分类器的鲁棒性以及知识蒸馏损失 $\mathcal{L}_{VA}$ 用于上下文信息注入，其公式为 $\mathcal{L}_{base} = \mathcal{L}_{seq} + \mathcal{L}_{VE} + \mathcal{L}_{VA}$。然而，基线方法只能预测词汇，这限制了模型学习细粒度子动作之间详细过渡和依赖关系的能力。因此，我们引入了一个层次子动作树机制，并利用它来实现视频与文本词汇之间的多层次跨模态对齐。

2) 细粒度子动作生成器: 为了解决现有数据集中缺乏细粒度标注的问题，我们建议利用大型语言模型（LLMs）来探索每个词汇表征中的潜在信息，为每个词汇表征的每个子动作生成详细且顺序的描述。详细信息见 Section III-C。

3) 分层子动作树构建器: 为了充分利用这些信息并实现手语的时间建模，我们设计了专门的提示来引导 LLM 生成与特定词汇表 $y_i \in Y$ 相关的子动

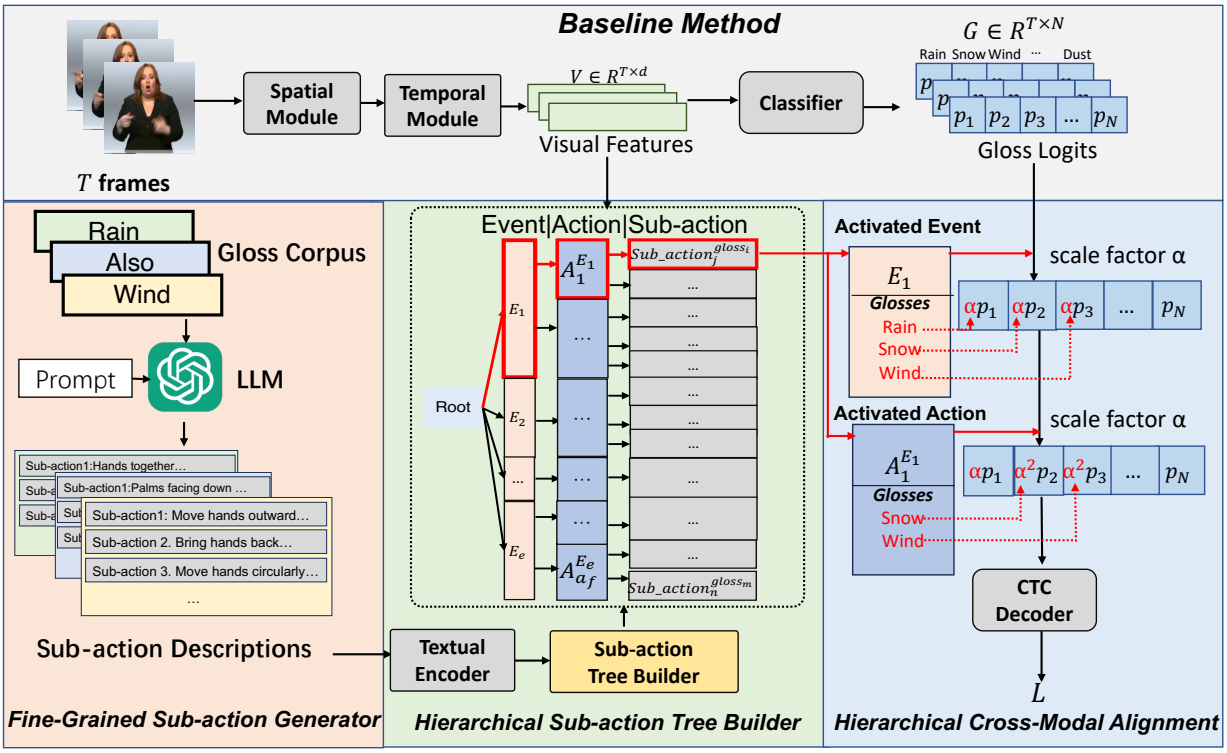

图 2. 我们提出的 HST-CSLR 框架。为了探索细粒度子动作信息，我们使用一个大语言模型为每个术语生成详细描述，并构建一个层次子动作树（HST）。应用最优路径搜索算法来整合子动作序列中的语义和时间信息，并通过激活的树节点增强视觉-文本一致性，以实现层级跨模态对齐。

作描述，其中 i 表示词汇表 i^{th}。这些提示旨在激发对相关子动作 $S_i = Sub\_action_k^i$ 的详细且上下文相关的描述（其中 k 是子动作为数）对于第 i 个词汇表 y_i，例如手势、表情和姿势。为了进一步组织并充分利用这些描述，我们应用了层次聚类方法。这产生了一个层次表示 $E = \{E_1, E_2, \ldots, E_e\}$，其中 e 表示顶层的聚类组数量。第二层包含一组较广泛的动作 $A = \{A_1^{E_1}, A_2^{E_1}, \ldots, A_{a_1}^{E_1}, \ldots, A_{a_e}^{E_e}\}$，其中上标 E_i 表明该二级节点属于父节点 E_i，其中 a_i 是 E_i 的中级动作数量，第三层包括叶节点。详情见 Section III-D。

4) 层次化的跨模态对齐: 为了有效地利用分层结构树（HST）中的路径信息，我们设计了一个层次交叉模态对齐模块。该模块专门用于充分利用层级路径信息，增强词义识别过程并提高整体模型性能。我们使用视觉特征 V 来评估不同层级的匹配程度，在分层结构树（HST）中选择一个最优路径。具体而言，基于 V 和 HST 中节点之间的对齐分数来确定最优路径 P^*。沿着选定的路径 P^*，我们激活与术语 y^i 对应的节点，并相应地更新术语 y^i 的 logits$\boldsymbol{p}^i$ 以细化识别过程。详情见 Section III-E。

I hope you can provide a motion sequence about how a certain word is expressed in German Sign Language (GSL). Keep in mind that I want to break down the process of expressing the word into a sequence of actions in chronological order. So each element in the sequence is a short action phrase without the need to be complete sentences. Just focus on the movements of the hands and head. If there are movements in both the hands and the head simultaneously, please describe at the same time. For example, you could give the action description of the word 'WETTER' in German Sign Language in the format of 'The sequence of how to express WETTER in German sign language is as follows. 1. XXX 2. XXX 3. XXX 4. XXX' with no other words (For example, do not add a note to the last line). Here is the word:

图 3. 一个详细提示的示例。

C. 细粒度子动作生成器

有效利用嵌入注释中的语义信息对于提升模型性能至关重要。我们假设利用大型语言模型（LLM）生成手语动作的文本描述可以显著改善视觉和文本模态之间的对齐，从而提高模型的整体性能。为了确保高质量的动作描述生成，我们采用 GPT-4 作为描述生成器，并精心设计提示以优化其输出。详细的提示如 Figure 3 所示。如 Figure 1 所示，在我们精心设计的提示引导下，LLM 能够为每个注释生成 4-5 个详细子动作描述，具体说明运动、动作和姿势等方面。

不可避免地，一些低质量的描述性语句，包括错误或无意义的词汇，在生成的子动作描述中出现。这些有缺陷的描述会对对齐过程产生明显的负面影响。为缓解这一问题，我们建议手动过滤并重新生成此类描述以消

除其负面影响。

D. 分层子动作树构建器

为了全面捕捉手语注释中的层次结构和语义关系，我们建议将子动作描述组织成树状表示。这种层级建模不仅能够有效地将视觉特征与文本描述在时间上对齐，还允许逐步利用细化的语义信息。通过构建分层子动作树（HST）并执行最优路径搜索，我们的目标是增强跨模态对齐并改进 CSLR。

1) 层次聚类在 HST 构造中的应用: 为了将子动作描述组织成结构化表示，我们构建了一个三层分层子动作树（HST）。首先，我们基于它们的语义相似性对所有子动作描述应用 K-means 聚类，创建 N_1 一级节点 $E = \{E_1, E_2, \ldots, E_{N_1}\}$。每个 E_i 包含语义相似的子动作描述。

为了进一步细化聚类，我们在每个一级节点 E_i 内执行第二轮聚类以形成二级节点 $A = \{A_1^{E_i}, A_2^{E_i}, \ldots, A_{a_i}^{E_i}\}$，其中 a_i 是 E_i 下的二级节点数量。在此步骤中，我们通过将原始聚类样本的后续陈述包含到聚类集中来确保时间一致性。最后，每个二级节点的内容被视为独立的叶节点，形成了 HST 的第三层。这个层次聚类过程组织子动作进入语义和时间上一致的组，结果形成一个三层树。

2) 最优路径搜索以实现视觉动作匹配: 为了使视觉特征与 HST 对齐，我们设计了一个最优路径搜索算法。设视觉特征为 $V \in \mathbb{R}^{T\times d}$，将其与从一级节点的每个节点 E_i 中使用文本编码器提取出的文本特征 $t_i^E \in \mathbb{R}^{1\times d}$ 进行比较。余弦相似度计算如下：

$$d_{V,t_i^E} = \cos(V, t_i^E), \tag{1}$$

$$p_i^E = \operatorname{argmax} d_{V,t_i^E}. \tag{2}$$

相似度最高的节点 p_i^E 被选中，然后对二级节点 $A_j^{E_i}$（E_i 的子节点）及其文本特征 $t_j^A \in \mathbb{R}^{1\times d}$ 重复相同的过程。最后，搜索范围扩展到叶节点（第三层），其中每个子动作的文本特征与 V 进行匹配，以在最细粒度级别上找到最佳对齐。结果路径 $p = \{p_i^E, p_j^A, p_k^S\}$ 将根连接到最佳匹配的叶节点。

E. 层次交叉模态对齐

1) 分层更新: 如何利用 HST 中的路径信息对于提升视觉和文本的对齐至关重要。我们设计了一种分层更新方法，以充分利用路径信息来辅助术语识别。具体来说，由于路径上的每个节点包含一系列陈述，我们将这些陈述所属的术语索引记录下来，并将细粒度的陈述与术语连接起来。然后我们通过乘以一个更新尺度 α 来突出分类器输出 $l_{origin} \in \mathcal{R}^{T\times N}$ 中的相应 logits，其中 N 是类别数量。值得注意的是，更新程度由分层路径内的出现次数决定。因此，对于叶节点陈述对应的术语，它应该被更新三次，即乘以 α^3。这一操作背后的原理是，在路径上多次出现的术语显然与视频片段有更高的对应度，因而应获得更高的关注。通过实施这种分层更新方法，我们充分利用了存储在路径中的信息，逐步加强视觉和文本之间的一致性，从而优化识别过程。在实践中，我们使用更新矩阵 $W \in \mathcal{R}^{T\times N}$ 来记录更新系数，并计算 W 和 l_{origin} 的哈达玛积以获得更新后的对数几率 $l_{updated}$：

$$l_{updated} = l_{origin} \odot W. \tag{3}$$

2) 跨模态对比对齐: 模型能否找到正确的层级路径是构建和更新我们的 HST 有效性的关键。然而，该路径可能在没有额外监督的情况下引入额外的噪声和错误的释义信息，这源于视觉模态与文本模态之间低质量的对齐。为了提高对齐质量，我们进一步引入了一个对比对齐损失。首先，我们采用 logitsl_{origin} 中的 argmax 来生成单个释义的伪标签 y^*。然后，我们选择每一层中与伪标签匹配的节点，即包含对应于 y^* 的陈述作为正样本，其他节点则作为负样本。每一层的对比损失定义如下：

$$\mathcal{L}_c = -\frac{1}{N_{positive}} \log \frac{\sum_{i\in p_{positive}} \exp(cos(v, t_{k,i}))}{\sum_{j\in p_{negtive}} \exp(cos(v, t_{k,j}))}, \tag{4}$$

其中，$p_{positive}$ 是正样本集，$p_{negtive}$ 是负样本集，k 是层数，$N_{positive}$ 是 $p_{positive}$ 的数量。

F. 训练和推理

1) 训练：: 目标函数形式如下，$\mathcal{L} = \mathcal{L}_{base} + \mathcal{L}_c$，其中 $\mathcal{L}_{base}$ 是我们基线采用的目标，而 $\mathcal{L}_c$ 是我们提出的对比对齐增强项。

2) 推理：: 在推理过程中，我们首先利用基线模型提取视觉特征并预测所有注释的初始 logits。然后根据视觉特征与每层树节点之间的对齐情况，在分层子动作树（HST）中搜索最优路径。使用选定的路径更新相应

的 logits 以细化预测结果。最后，将更新后的 logits 用作注释识别的最终预测结果。

IV. 实验

A. 实现细节

遵循之前的 [15], [16] 方法，我们使用四个数据集：PHOENIX-2014 [9], PHOENIX-2014T [10], CSL-Daily [11] 和 Sign Language Gesture [12]。对于评估指标，我们也遵循之前的工作 CorrNet [5]，这是我们基准线，采用词错误率（WER）。为了公平比较，我们遵循之前工作的结构 CorrNet [5]，该工作采用了 ResNet-18 作为主干网络。为了编码描述，我们使用 BERT [17] 作为文本编码器。我们的模型训练了 80 个周期，初始学习率为 0.001，在第 40 和 60 个周期时将学习率除以 5。采用了 Adam 优化器，权重衰减为 0.001，批量大小为 2。我们在分层更新中设置的更新比例 α 为 1.5。

表 I
PHOENIX-2014 [9]，PHOENIX-2014T [10] 的比较。

模型	凤凰-2014		凤凰-2014T	
	Dev ↓	Test ↓	Dev ↓	Test ↓
SubUNets [18]	40.8	40.7	-	-
Staged-Opt [19]	39.4	38.7	-	-
Align-iOpt [20]	37.1	36.7	-	-
SFL [21]	26.2	26.8	25.1	26.1
VAC [4]	21.2	22.3	-	-
STMC [2]	21.1	20.7	19.6	21.0
SMKD [3]	20.8	21.0	20.8	22.4
C^2SLR [7]	20.5	20.4	20.2	20.4
CVT-SLR [14]	19.8	20.1	19.4	20.3
CorrNet [5]	18.8	19.4	18.9	20.5
Two-Stream SLR [6]	18.4	18.8	17.7	19.3
TCNet [16]	18.1	18.9	18.3	19.4
CSGC [15]	18.1	19.0	**17.2**	19.5
Ours	**17.9**	**18.2**	17.4	**19.1**

B. 与最新技术的比较

我们将我们的结果与现有的最先进方法进行了比较，结果显示在 Tables I to III 中。我们在四个数据集上的实验表明了我们模型在德语、中文和英语上的泛化能力，并且对视频和图像都表现有效。得益于我们的分层更新过程，我们获得了性能提升，并超越了之前的最先进方法，在 PHOENIX-2014 [9] 的 Dev/Test 集上获得 0.2%/0.6% 的性能提升，在 PHOENIX-2014T [10] 的测试集上获得了 0.2% 的提升，这表明我们的细粒度分层子动作树和更新设计可以有效地提供全面的上下文和时间建模信息。然而，PHOENIX-2014T [10] 的开发集上的改进是有限的。我们认为这是由于这些数据集中获取高质量且细粒度描述的固有难度，因为它们包含噪音和模糊不清的内容。我们将提供更精确的细粒度描述以提高性能作为进一步研究的方向。

表 II
使用 WER(%) 报告的 CSL-每日比较。

	SEN [22]	CorrNet [5]	TCNet [16]	Ours
Dev	31.1	30.6	29.7	**27.5**
Test	30.7	30.1	29.3	**27.4**

表 III
手势语比较 [12]。

Model	Accuracy (%)
Baseline	97.28
Ours	**99.85**

C. 消融研究

所有消融研究都在 PHOENIX-2014T 数据集上进行，并以字错率（%）报告。[10]

表 IV
不同输入下的性能。

$\mathcal{L}_{\text{Seq}}$ 计算	解码器 输入	WER (%)	
		Dev	Test
w/ l_{origin}	l_{origin}	18.9	20.5
	$l_{updated}$	-	-
w/ $l_{updated}$	l_{origin}	18.2	19.4
	$l_{updated}$	**17.4**	**19.0**

分层更新的影响：我们分别使用原始 logitsl_{origin} 和我们的更新后的 logits$l_{updated}$ 作为训练中 CTC 损失 $\mathcal{L}_{\text{Seq}}$ 的输入，并将 l_{origin} 和 $l_{updated}$ 作为 CTC 解码器的输入来计算 WER。如表 IV 所示，与 l_{origin} 相比，$l_{updated}$ 在 Dev/Test 分割上分别展示了 0.8%/0.4%的性能提升，无论是作为训练过程中的 $\mathcal{L}_{\text{Seq}}$ 输入还是测试过程中 CTC 解码器的输入。

表 V
不同更新尺度 α 在开发集上的比较。

α	1.25	1.5	1.75	2.0
WER (%)	18.4	**17.4**	18.6	19.5

更新尺度的影响 α：我们研究了不同更新规模对性能的影响，如表 V 所示。将 α 设为 1.5 获得了在 PHOENIX-2014T 数据集上的最佳表现，得分为 17.4。过小的 α 对预测过程的影响较小，并且无法显著影响性能，而过大的 α 则可能阻碍模型的鲁棒性并导致性能下降。

表 VI
树深度（聚类参数）的消融研究。

	CorrNet(baseline)	1	2	3
Dev	18.9	18.2	17.8	**17.4**
Test	20.5	20.1	19.5	**19.0**

树结构的影响：我们分析了不同树结构的影响，这主要集中在树的深度上。性能显示在表 VI 中。如所示，采用一层或两层的树结构不足以提供细粒度的视觉和时间信息以改善性能。使用三层来构建子动作树，我们在 Dev/Test 上获得了 1.5%/1.5% 的性能提升，表明细粒度的子动作可以为手势识别明确地提供有价值的信息。

表 VII
正负样本

Model	Dev	Test
w/o	17.5	19.6
w/	**17.4**	**19.0**

正负样本的影响：树中包含与此伪标签相对应的描述句子的节点被视为**正面的**(正)，而其余节点则被认为是**负的**(负)。如 Table VII 所示，我们可以利用正负样本提升性能。

表 VIII
HST 和跨模态对齐的消融研究。

	CorrNet(baseline)	baseline + HST	baseline + $\mathcal{L}_c$	our
Dev	18.9	17.6	18.1	**17.4**
Test	20.5	19.6	19.8	**19.0**

跨模态对齐的影响：跨模型对齐的影响显示在 Table VIII 中。如所示，通过应用分层子动作树和更新，与基线相比，我们获得了 0.7%/0.4% 的性能提升。进一步引入跨模态对齐可以再获得 0.2%/0.6% 的性能提升，这表明引入这种跨模态对齐可以有效地对齐视觉和文本特征，从而增强光泽识别能力。

表 IX
过滤方法

Method	Dev	Test
LLMs	17.9	19.6
Manual	17.4	19.0

手动过滤的影响：如图 Table IX 所示，手动过滤优于大语言模型自我过滤，但大语言模型自我过滤的整体性能仍超越大多数现有方法。

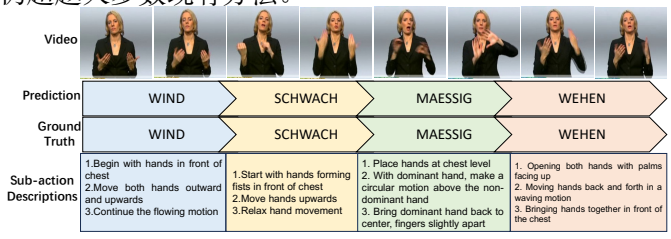

图 4. 视觉示例。

视觉示例：Figure 4 显示了一个视觉示例，包括来自大语言模型的子操作描述以及我们的预测与真实情况之间的比较。我们预测与真实情况之间的一致性显示了我们方法的有效性。

V. 结论

在本文中，为了应对连续手语视频缺乏细粒度理解的问题，我们提出了面向 CSLR 的分层子动作树 (HST-CSLR)，该方法高效地将来自大型语言模型（LLMs）的术语特定知识与视觉表示学习结合起来。通过构建分层子动作树（HST），我们的方法在对齐视觉和文本模态的同时利用结构化表示来降低复杂度。此外，我们引入对比对齐增强以更好地弥合跨模态差距。实验表明，HST-CSLR 达到了最先进的或具有竞争力的性能，显示了其在改进 CSLR 方面的有效性。

VI. 致谢

本工作得到了国家自然科学基金 62372014 和北京市自然科学基金 4252040 的支持。

参考文献

[1] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006.

[2] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li, "Spatial-temporal multi-cue network for continuous sign language recognition," in *Proceedings of the AAAI conference on artificial intelligence*, 2020.

[3] Aiming Hao, Yuecong Min, and Xilin Chen, "Self-mutual distillation learning for continuous sign language recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[4] Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen, "Visual alignment constraint for continuous sign language recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[5] Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng, "Continuous sign language recognition with correlation network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[6] Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak, "Two-stream network for sign language recognition and translation," *Advances in Neural Information Processing Systems*, 2022.

[7] Ronglai Zuo and Brian Mak, "C2slr: Consistency-enhanced continuous sign language recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[8] Peiqi Jiao, Yuecong Min, Yanan Li, Xiaotao Wang, Lei Lei, and Xilin Chen, "Cosign: Exploring co-occurrence signals in skeleton-based continuous sign language recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[9] Oscar Koller, Jens Forster, and Hermann Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, 2015.

[10] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden, "Neural sign language translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[11] Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li, "Improving sign language translation with monolingual data by sign back-translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] Aadershi Mohan, Daivik Mohan, Satvik Vats, Vikrant Sharma, and Vinay Kukreja, "Classification of sign language gestures using cnn with adam optimizer," in *2024 2nd International Conference on Disruptive Technologies (ICDT)*. IEEE, 2024.

[13] Huaiwen Zhang, Zihang Guo, Yang Yang, Xin Liu, and De Hu, "C2st: Cross-modal contextualized sequence transduction for continuous sign language recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[14] Jiangbin Zheng, Yile Wang, Cheng Tan, Siyuan Li, Ge Wang, Jun Xia, Yidong Chen, and Stan Z Li, "Cvt-slr: Contrastive visual-textual transformation for sign language recognition with variational alignment," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023.

[15] Qi Rao, Ke Sun, Xiaohan Wang, Qi Wang, and Bang Zhang, "Cross-sentence gloss consistency for continuous sign language recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[16] Hui Lu, Albert Ali Salah, and Ronald Poppe, "Tcnet: Continuous sign language recognition from trajectories and correlated regions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, vol. 38.

[17] Jacob Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[19] Runpeng Cui, Hu Liu, and Changshui Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[20] Junfu Pu, Wengang Zhou, and Houqiang Li, "Iterative alignment network for continuous sign language recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[21] Zhe Niu and Brian Mak, "Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*. Springer, 2020.

[22] Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng, "Self-emphasizing network for continuous sign language recognition," in *Thirty-seventh AAAI conference on artificial intelligence*, 2023.