Universal Access in the Information Society (2026) 25:45
https://doi.org/10.1007/s10209-026-01310-1

LONG PAPER

Signbuddy: from sign language research to scalable co-created solutions

Toon Vandendriessche[1] · Caro Brosens[2] · Hannes De Durpel[2] · Mathieu De Coster[1] · Joni Dambre[1]

Received: 6 March 2025 / Accepted: 8 January 2026

Abstract

This paper presents SignBuddy, the result of ongoing co-created sign language processing research. Mostsign language processing research is performed by hearing, non-signing researchers. Even though co-creation efforts have recently increased, technical research still often fails to mention if (and how) co-creation was involved in the research process. SignBuddy is a co-created research tool developed through apartnership between the Flemish Sign Language Centre, a deaf-led organisation, and Ghent University. While respecting elemental concepts of co-creation - i.e. (i) defining common goals and (ii) building a formal and sustainable relationship between users/consumers and researchers/developers and respectingthe five lessons in co-creation - the platform successfully supported the development of **the first fully scalable sign-to-text dictionary search system, built into the Flemish Sign Language–Dutch onlinedictionary**. SignBuddy functions as a crowdsourcing interface for *in-the-wild* collection of modelevaluation data, gathering example queries for quantitative performance analysis and user feedback forqualitative assessment. This human evaluation allows us to shape the application based on the end-users'needs. Addressing the need for models that support large dictionaries (over ten thousand signs), we propose a scalable one-shot sign language recognition method and achieve state-of-the-art results. Beyond the co-created application itself, this work provides insights into the co-creation process - clarifying roles, shared goals, and responsibilities - and offers conclusions to guide future co-created sign language processing research.

Keywords Sign language · Natural language processing · Computer vision · Assistive technologies

1 Introduction

The last decade has seen a steady increase in published research related to *sign language processing* (SLP), i.e., the development of language technology for sign languages [3, 4]. Unfortunately, most of this research was done largely or exclusively by hearing, non-signing researchers and with a strong focus on technological achievements. Many of the proposed solutions are consequently not suitable for or tailored to their targeted users, because they ignore fundamental aspects of the richness and diversity of sign languages or are restricted to usage scenarios that are not useful for or acceptable to their targeted users [5]. As a result, the developed technologies rarely find their way to practical applications. In fact, due to past disappointments, many *deaf and hard of hearing* (DHH) persons have even become wary of sign language technology altogether [6, 7].

The participation of DHH and signing researchers in the development of SLP is essential for avoiding the shortcomings that arise when technologies are developed without the perspectives of those who use them. Yet the number of DHH professionals with technical training in machine learning remains limited. In the United States, institutions such as Gallaudet University—where American Sign Language (ASL) is the primary language—provide more extensive

✉ Toon Vandendriessche
toon.vandendriessche@ugent.be

Caro Brosens
caro.brosens@vgtc.be

Hannes De Durpel
hannes.de.durpel@vgtc.be

Mathieu De Coster
mathieu.decoster@ugent.be

Joni Dambre
joni.dambre@ugent.be

[1] IDLab-AIRO, Ghent University - imec, Technologiepark-Zwijnaarde 126, 9052 Zwijnaarde, Belgium

[2] VGTC, Dorpsstraat 43a, 9052 Zwijnaarde, Belgium

Published online: 20 February 2026

academic pathways for DHH students. In contrast, across much of Europe, DHH researchers with the relevant expertise are far less prevalent. Although this scarcity makes it challenging to recruit technically trained DHH collaborators, their involvement in sign language research is widely recognised as indispensable [1, 5, 8].

In response to these constraints, several recent projects led primarily by hearing researchers have adopted co-creation methodologies (e.g., SignON [9], EASIER [10], CoCoS [11], SignGPT [12]). However, as both Lepp et al. [1] and De Meulder et al.[2] observe, many of these efforts have struggled to move beyond superficial participation and have rarely realised the structural rebalancing that co-creation presupposes. According to Lepp et al., genuinely co-created solutions require a shared goal and a formal, sustainable partnership, yet such partnerships often remain aspirational rather than fully operationalised. Similarly, De Meulder et al. argue that co-creation demands a fundamental reconfiguration of who is recognised as a designer and decision-maker—something that most existing SLP projects have not successfully achieved.

Ideally, co-creation begins with jointly identifying needs and priorities and continues through cycles of design, evaluation, and refinement. When implemented in full, this process embeds community perspectives into the resulting technology, thereby reducing the risk of developing tools that are misaligned with real-world needs [13]. Yet the limited success of prior co-creation attempts underscores that achieving this vision requires more than participatory activities: it demands a redistribution of authority, sustained trust, and structural support for DHH leadership [2].

Acknowledging both the challenges and benefits of co-creation, this paper presents collaborative research between (hearing) researchers at Ghent University (UGent, Belgium) and (deaf or hard of hearing) researchers at the Flemish Sign Language Center (VGTC, Belgium), a deaf-led expertise centre committed to advancing the knowledge of and about Flemish Sign Language (Vlaamse Gebarentaal, VGT) through research, education, and community engagement. As one of its core activities, VGTC maintains and expands the publicly accessible VGT-Dutch online dictionary [14].

Our partnership was guided by the core conviction that, given the current state of technology related to SLP, it must be possible to develop applications that are truly useful to the Flemish signing community. By working closely together, we ensured that the expertise and perspectives of (DHH) signers remained central to the research process—safeguarding not only functional utility but also linguistic authenticity and cultural relevance. The targeted application of our joint research is a robust and scalable video based dictionary search system that allows to look up the meaning of a sign in the entire online VGT-Dutch dictionary, both for currently documented signs and for signs that are yet to be added in the future. To analyse how well the proposed application will work, we collect and analyse usage data with a publicly available tool that we call *SignBuddy*.[1] The contributions of this paper are threefold:

1. We report on our *co-creation process* and the many ways in which sign language knowledge and signers' perspectives were embedded into our research decisions at all stages.
2. We present the core contribution in our envisaged path towards the targeted application: the targeted collection of user data that allows us to *evaluate the scalability of our approach* and future improvements, and to evaluate whether our model's outputs are based on relevant properties of signs and to identify common failure cases.
3. We provide an initial *analysis* of the data that has been collected up to the time of writing and identify promising directions for improvement, both to the user interface and to the underlying technical model.

The remainder of this paper is organised as follows: Sect. 2 gives an overview of related work, Sect. 3 details the ideation of SignBuddy and its requirements, Sect. 4 illustrates the SignBuddy application design arising from this, Sect. 5 goes over our analyses of the data collected through the SignBuddy platform, Sect. 6 gives some insight to the transition from SignBuddy to the first publically available sign-to-text dictionary, and Sect. 7 lists our insights for future work on SignBuddy and sign language dictionary search functionality in general.

2 Related work

2.1 The data challenge in sign language processing

Sign languages are complex languages with rich vocabularies and a lot of variation in the way they are expressed. Most sign languages, however, are less standardised than written or spoken languages. They have fewer lexemes that are fully conventional in form and meaning compared to most spoken languages, but allow for greater creativity in (re)combining meaningful components to express new meanings or nuances. This is further enhanced by the fact that, unlike spoken languages, sign languages also exhibit simultaneity, which is the concurrent execution of different lexical and morphological elements. Not nearly all of the meaningful building blocks or potential combinations and meanings have been documented.

[1] https://woordenboek.vlaamsegebarentaal.be/signBuddy.

Despite this, the success of artificial intelligence in processing written and spoken languages has prompted research into sign language processing as well. Sign language linguists and computer scientists working in computer vision and natural language processing are pursuing the automatic analysis and translation of sign languages, using powerful deep neural networks. These should in principle be able to achieve good performance, but typically need a lot of high-quality training data. Unfortunately, in comparison to many written and spoken languages, sign language data that is available (in compliance with the General Data Protection Regulation (GDPR)), properly annotated, and suitable for research purposes is extremely scarce.[2] As data availability is the main bottleneck for high-quality SLP, the most notable research efforts today focus on data-efficient techniques such as learning from unlabeled data [15, 16], transfer learning from related data sets [17, 18] or preprocessing to simplify the task [19, 20].

2.2 Isolated sign language recognition

Isolated sign language recognition (ISLR) focuses solely on recognising (classifying) individual lexical items. Glosses can serve as unique and interpretable labels for this classification task, as they are a written representation of signs. Glosses can denote individual lexical items (e.g., CAR, BRIDGE) or depict multi-sign constructions (e.g., car-drives-across-bridge).

Gloss-based ISLR has recently been used to mitigate the data bottleneck for sign language translation [21]. In this approach, the signs in continuous signing are identified by sliding a windowed ISLR model over the video, after which the resulting sequence of glosses is processed further by a (text-based) translation model. While this offered promising results, the use of glosses in sign language translation (SLT) is a topic of ongoing debate [4, 22], because translation requires capturing linguistic and semantic nuances beyond the gloss level. A single written word, i.e. gloss, cannot capture the full semantic and grammatical richness of a sign. Therefore, glosses are never meant to be a translation of a sign. In the remainder of this section, we focus on ISLR, since that is the type of SLP used in SignBuddy.

The input videos for ISLR can be processed in several ways [17, 23]. Recent advancements [24, 25] have demonstrated a positive impact using pose estimation models, such as MediaPipe Holistic [26] (henceforth MediaPipe) and OpenPose [27]. These models transform input videos into sequences of skeletal representations, capturing "keypoints" or "landmarks" of the human pose in 2D or 3D Cartesian coordinates. Removing information about a person's appearance allows ISLR models to focus solely on the structural aspect of sign videos: how people move their arms, hands, and face. This enhances the generalisability to downstream tasks, while also enabling anonymity. Overall, the adoption of MediaPipe has significantly advanced the state-of-the-art of ISLR.

However, there is still room for improvement in preprocessing. Although the keypoint estimator is generally accurate, MediaPipe sometimes produces erroneous keypoint predictions when the two hands interact. Since this interaction is elemental to sign language, crucial information is lost. Moryossef et al. [19] argued that this tool is not directly applicable to fine-grained tasks such as sign language recognition. However, recent Kaggle competitions [28, 29] based on keypoint estimation using MediaPipe present a different perspective, showing promising results in SLR using keypoint estimation. A key component appears to be the addition of a frame embedder—an architectural component that generates per-frame latent representations (also called embeddings),[3] without considering temporal context—that is not present in the work by Moryossef et al. [19], but present in all top Kaggle competition entries. This frame embedder allows the network to learn the non-linear relationships between keypoints [20].

Besides the preprocessing of sign language videos, the architecture of the models also has a large impact. Until 2020, deep learning approaches to ISLR primarily used variations of Recurrent Neural Networks [30–32]. The common factor of these models is that they are proficient at handling sequential data and dealing with temporal dependencies between different poses. The introduction of transformers [33] in 2017 initiated a paradigm shift. The combination of keypoints and attention leads to powerful models for ISLR: the top scoring method on the Kaggle ASL ISLR competition [28] achieved 89.3% test set accuracy on 250 sign categories using keypoint data and attention.

Even with the addition of preprocessing techniques like keypoint extraction, the lack of training data still poses a hard bottleneck on the supported vocabulary of classification approaches based on supervised learning. For each sign, they need a sufficiently large and approximately equal number of examples for training to allow for the variability in sign execution. As already mentioned, sign languages are complex and evolving languages with large vocabularies, which motivates the need for more flexible ISLR techniques that can easily be adapted to include additional signs.

[2] Admittedly, this problem is not unique to only sign languages, but it does, however, apply to all sign languages.

[3] An embedding is a representation of the relevant properties of the input, in this case the sequence of keypoints extracted from a single sign video, into a vector of numbers. A good embedding model should maximally represent the relevant information while ignoring information that is not relevant to the targeted task.

Fei-Fei et al. [34] argued that "one can take advantage of knowledge coming from previously learned categories, no matter how different these categories might be." This insight led to the introduction of few-shot learning, where fewer examples per category are required. Wang et al. [35], for example, leverage multiple examples of one sign to perform K-means clustering and a custom matching algorithm. For some sign languages, a dictionary is available. Resources like sign language dictionaries—extensive collections of one or more example videos of a given sign—can also be used for few-shot learning. In the case of VGT, the dictionary contains exactly one example per unique documented sign. De Coster and Dambre [36] employed pretrained model embeddings in a Euclidean distance-based vector search, in essence performing one-shot learning to recognise signs in the VGT dictionary [14]. However, the model's performance on dictionary lookup was still quite poor, as will be shown in Sect. 4.2. This limited performance can primarily be attributed to the Zipfian class distribution in the pretraining dataset, which was extracted from the VGT corpus [37]. Due to this imbalance, the model has not been exposed to a wide enough variety of signs during training and therefore is not able to distinguish enough properties of signs to perform one-shot SLR. In addition, the examples used in the pretraining dataset were extracted from continuous signing rather than isolated signs, which poses a domain mismatch.

Table 1 Online, publicly available sign language dictionaries, their number of signs (size) and sign-to-text (S2T) search modality

Sign language	Size	S2T
No sign-to-text search available		
ASL (American)	40,000	None
LSF (French)	27,025	None
BSL (British)	21,000	None
KSL (Kenyan)	8926	None
GESL (Georgian)	8296	None
NTS (Norwegian)	6500	None
DGS (German)	5509	None
AUSLAN (Australian)	4912	None
FinSSL (Finl.-Swed.)	3035	None
TSL (Taiwan)	2194	None
HSL (Hawai'i)	1500	None
Search by using sign parameters (SP)		
LESCO (Costa Rican)	1041	1 SP
Libras (Brazilian)	3093	1 SP
NZSL (New Zealand)	4500	2 SPs
HKSL (Hong Kong)	7346	2 SPs
DTS (Danish)	2250	3 SPs
PJM (Polish)	3476	3 SPs
NGT (Dutch)	1600	5 SPs
SSL (Swedish)	21,734	6 SPs
Search by sign language processing (SLP)		
LSFB (French-Belgian)	11,859	700 signs
VGT (Flemish)	11,248	2 SPs + full dictionary

The URL to the dictionary is embedded within the sign language name

2.3 Adoption of SLR tools in practical applications

While the scientific literature has reported significant advancements in ISLR performance, its integration in tools that offer an added value to DHH communities unfortunately remains limited. Examples of existing applications include the support of *sign language acquisition* and *dictionary search systems*. While these applications serve distinct purposes, both rely on ISLR.

Sign language acquisition tools demonstrate how ISLR can support real learning scenarios by providing feedback on whether a user reproduces a target sign accurately. Several mobile tools illustrate the feasibility of deploying ISLR in everyday settings. PopSign [38], for instance, helps hearing parents of DHH children learn frequently used ASL signs through a gamified interface, enabled by a diverse smartphone-recorded dataset.[4] Similarly, Zhang et al. [39] propose a smartphone-based system for learning 1,000 Chinese signs, demonstrating the potential scalability of lightweight setups.

Other approaches explore more structured environments that require specialised hardware. SMILE [40] places both DHH and hearing children in a virtual-reality setting to learn a limited set of ASL signs alongside academic content, while CopyCat [41, 42] evolved from glove-based sensing to depth-camera tracking and pose estimation. These systems highlight how controlled environments can support richer feedback but at the cost of accessibility.

Across all these efforts, a central limitation remains: current tools rely on fixed, tightly curated vocabularies defined during data collection. As a result, learners can only practise a restricted set of signs, even though real learning needs vary widely across individuals and contexts. Broadening these tools to support larger or customisable vocabularies would substantially increase their usefulness—yet doing so requires scalable sign-to-text capabilities, a challenge we address in this work.

Online sign language dictionaries traditionally provide a search functionality to look up the sign or signs that best correspond to a written word (text-to-sign search), examples are listed in Table 1. Some noteworthy examples include SignASL (ASL Sign Language Dictionary), SignBSL (BSL Sign Language Dictionary) and the VGT dictionary [14, 43], comprising over 40,000, 21,000, and 11,000 signs, respectively. Although these resources are designed for the signing community, sign language dictionaries are often used by second language learners, as text-to-sign search

[4] A total of 94,477 samples covering 250 signs and recorded by 21 different signers [28].

is particularly relevant for this user group. Nevertheless, improving access to sign language learning indirectly benefits the DHH community. In contrast, sign-to-text search—i.e. performing a sign to look up its textual translation—is more relevant for the signing community, but remains far more uncommon.

The most common way to achieve this bidirectionality today is by searching the parameters of a sign, the phonological building blocks of signs, for example an overview of all signs with a certain handshape or on a certain location on the body. The dictionaries for Swedish Sign Language (SSL), New-Zealand Sign Language (NZSL), Sign Language of the Netherlands (NGT), French Belgian Sign Language (LSFB) and Flemish Sign Language (VGT) all offer this search function. The existence of these search functions illustrates the need and want of the community for bidirectionality, however in its current form it appears not efficient enough. While this narrows down the search considerably, it does not allow the user to search for one specific form. A notable effort to automate searches by Fink et al. [44] enables users to perform such searches. Unfortunately, the underlying model for this example is based on supervised learning. This means that the tool has been trained to address a fixed and constrained vocabulary which can not be extended to new signs without collecting training examples for these signs and retraining the model. Concretely, the lookup functionality of Fink et al. [44] supports a vocabulary of 700 signs—merely a fraction of those found in sign language dictionaries. Since datasets with sufficient samples of (much) larger vocabularies are not available, it remains difficult to assess whether any of the approaches mentioned in Sect. 2.2 can be scaled up to capture the full lexical richness of sign language.

2.4 The VGT dictionary

The Flemish Sign Language (VGT) dictionary [14] is the primary lexicographic resource for Flemish Sign Language. Through its continual development in close collaboration with the Flemish signing community, it plays a central role in preserving, documenting, and teaching the language, while reflecting its contemporary lexical and regional diversity. The dictionary's origins trace back to smaller lexicographic projects in Flanders initiated in 1999. In 2004, the Flemish universities UGent and KU Leuven released the first public version, which was subsequently transferred to VGTC in 2012, who have managed it ever since.

At the time of writing, the VGT dictionary comprises 11,248 unique signs. However, VGTC manages over 20,000 Flemish Sign Language entries, including those in the dictionary, in SignBank [45]—an electronic database developed to compile and manage lexicographic data for sign languages. In SignBank, each sign entry is represented by a unique gloss, consisting of a word and a letter suffix. The word corresponds to a single concept, while the letter suffix distinguishes variants of the same concept—i.e. synonyms. For example WONEN-A[5] refers to the concept of *'to live'*, just like WONEN-B, WONEN-D, WONEN-F, WONEN-H, and WONEN-I. These synonyms sometimes correspond to the different provinces in Flanders. Granted that glosses are practical for managing signs, they do not always provide a good indication of the full meaning of a sign. For example MUSEUM-J can not only be translated to *museum* but also *tentoonstellen* (to exhibit), *tentoonstelling* (exhibition), *expo* (expo), *salon* (salon), and *beurs* (fair). A user could find this particular sign by searching for any of these listed translations—illustrating the text-to-sign search functionality.

From a user perspective, text-to-sign search is primarily relevant for non-signers and novice signers, which is reflected in the current user base: most active users of the VGT dictionary are hearing VGT learners. Incorporating more specialised vocabulary and explanatory elements—such as definitions and example sentences—would increase the dictionary's relevance for the DHH community. At present, the VGT dictionary functions only as a translation resource, but VGTC is working towards developing such explanatory dictionary. This expanded functionality could particularly benefit DHH users with lower proficiency in Dutch, allowing them to consult the dictionary in their native language.

It is important to note that the VGT dictionary is not a static set of unique signs, but an ever-expanding collection maintained through several curation mechanisms. Although VGTC manages the dictionary, it does not unilaterally determine which signs are selected from SignBank. While corpus research informs the selection process where possible, VGTC also relies on community involvement to collect and verify signs. The primary mechanism is the *Gebarencommissie* (signing committee), which is a group of five highly proficient signers—one from each of the five provinces in Flanders—that meets once a month to discuss signs. Another is the *Gebarometer* (Sign-o-meter), a daily email blast to a group of about 65 deaf signers, which surveys one sign a day. Finally, *targeted lacuna projects* focus on domain-specific vocabulary, discussed with signers who are active in the relevant field. These combined initiatives make the VGT dictionary not only a highly reliable resource but also an ever-evolving representation of the Flemish signing community.

Additionally, the VGT dictionary was among the first to support a form of sign-to-text search [46]. In 2004, users could search for sign translations using SignWriting [47].

[5] The URL to the videos that corresponds to the glosses is embedded into the gloss name.

Although revolutionary at the time, the feature saw limited uptake, as few users were familiar with SignWriting. In 2017, this led to a replacement with a parameter-based system, in which users search for signs by selecting handshape(s) from a list or indicating the articulatory location on a pictogram of the human body. While this parameter-based approach increased accessibility, it does not constitute a complete sign-to-text search, highlighting the need for a more direct and intuitive method.

In a genuine sign-to-text search, the user performs a sign in front of his computer and retrieves the relevant translation in Dutch. Developing a method capable of searching the entire dictionary is non-trivial for several reasons. First of all, (i) the method must be able to handle any given unique sign, which is represented by a unique gloss. Second, (ii) it must efficiently and robustly search through very large vocabularies. Finally, (iii) the dictionary's continuous evolution necessitates a flexible solution that does not require retraining after every addition. Through our collaboration, the VGT dictionary now supports such genuine sign-to-text search. The technical details of this implementation, however, lie beyond the scope of this paper, which focuses instead on the development process, collaboration and collected data.

3 The ideation of SignBuddy

This section outlines the conceptual foundations that informed the development of SignBuddy. Although some elements of the technical implementation are briefly referenced, a full formal description of the system is provided later in Sect. 4. We begin by presenting the co-creation framework that underpins the project in Sect. 3.1, establishing the collaborative principles that shape all subsequent design choices. Building on this foundation, we articulate the long-term objectives of the platform in Sect. 3.2 and then describe the intermediate steps that function as stepping stones toward these goals in Sect. 3.3.

3.1 Co-creation

While the works cited in Sect. 2.2 represent significant technical advancements in isolated sign language recognition—effectively creating stepping stones for the technical aspect of this work—few studies explicitly acknowledge the involvement of the DHH community (if they were involved at all) [1]. SignBuddy shifts this paradigm and implements a co-creation process. The recent theoretical exploration by Lepp et al. [1] describes how two concepts are essential in co-creation: (i) the existence of a common goal between users/consumers and researchers/developers, and (ii) a formal and sustainable relationship between those two groups. De Meulder et al. [2] offer an alternative perspective on co-creation, arguing that *"co-creation should not be reduced to mere participation, but understood as a reconfiguration of who is recognised as a designer and decision-maker."* This aligns more with our vision of co-creation—a collaboration between partners with equal status—and that is why we choose to follow this definition.

Building on this perspective, the expertise- and experience-driven work byDe Meulder et al. [2] distills five lessons from previous co-creation efforts in the context of SLP which we implemented in the SignBuddy co-creation process:

- *Lesson 1:* recognise and resource DHH partners' invisible labour;
- *Lesson 2:* manage expectations via accessible science communication;
- *Lesson 3:* "crip" co-creation by dismantling structural ableism;
- *Lesson 4:* diversify participatory methods to address co-creation fatigue and intersectionality;
- *Lesson 5:* redistribute power through DHH leadership.

In SignBuddy, DHH researchers from VGTC (C. Brosens, H. De Durpel) and hearing researchers from UGent (T. Vandendriessche, M. De Coster, J. Dambre) enjoy equal status. From the outset, SignBuddy was envisioned by researchers from VGTC—representing the DHH community—and UGent who (i) shared a common goal: developing sign-search functionality for the VGT–Dutch dictionary. Moreover, (ii) a formal and sustainable relationship between the partners has long been in place, as UGent and VGTC maintain ongoing collaborations beyond the SignBuddy project. The SignBuddy project is deaf-led, with VGTC serving as the primary partner in the consortium, directly employing *Lesson 5*.

The partnership has always been deliberately structured. Most decisions regarding the application, planning, and data collection protocol were made jointly, while purely technical choices—such as those concerning neural network architectures—were handled by UGent. These joint decisions were facilitated through recurrent meetings, aligning with *Lesson 3*, which ensured that both partners remained informed, coordinated, and able to contribute according to their working capacities. This structure avoided last-minute decision-making and supported a sustainable, inclusive workflow.

The SignBuddy project ensured that the broader DHH community was actively informed and engaged throughout its development, directly putting *Lesson 2* into practice. All information on the platform—including the project goals, workflow, privacy details, and data-collection protocol—was provided in both VGT and Dutch, ensuring accessible communication for all users. These materials also explain

how artificial intelligence is used in the system. By clarifying that the dictionary videos alone are insufficient to train the model and by openly requesting user feedback, the project transparently communicates the limitations of current technology. In doing so, it actively manages expectations, fulfilling the second component of *Lesson 2*.

By making the platform publicly accessible, we also mitigated the risks highlighted in *Lesson 4*, namely co-creation fatigue and issues of intersectionality. The flexible recruitment of participants—through both personal networks and visitors of the VGT-dictionary webpage and its call to action—resulted in a diverse set of contributions. This approach reduced pressure on the DHH researchers' personal networks and helped prevent the repeated overburdening of the same segment of the DHH community.

The success of SignBuddy depended not only on active collaboration but also on the complementary contributions of both partners. UGent delivered the technical components, but VGTC carried out labour that is less visible yet even more essential: consulting with the DHH community, translating project materials into VGT, making culturally appropriate design choices, and providing a trusted point of contact for participants. These tasks require sustained community engagement and are often informally assigned to DHH partners without being fully recognised or resourced. In SignBuddy, we explicitly acknowledged and supported this work, directly addressing a core concern of *Lesson 1* and ensuring that DHH partners' community-facing labour was treated as a formal and valued part of the project.

Taken together, the implementation of all five lessons shows that co-creation in SLP requires more than participation: it requires structural commitments to DHH leadership, transparent communication, shared decision-making, and explicit recognition of community-facing labour. The development of SignBuddy illustrates how these commitments can be realised in practice, and leads to a valuable end-result. The following sections give a structured overview of the collaboration itself.

3.2 From stakeholder needs to the primary research goal

As already mentioned, our focus was to maximally leverage previous research experience from the SignON project to the short-term benefit of the Flemish signing community. The goal of our collaboration, the creation of a video-based sign-to-text search functionality for the VGT online dictionary, was obtained by aligning the stakeholder priorities with the TRL (technology readiness level) and feasibility of potential SLP applications. It was identified as *desirable* by VGTC, supported by their regular consultation of DHH community and knowledge of the usefulness of such a tool for VGT learners and as *feasible* by UGent. Using these two markers, we set grounded *requirements*.

Desirability: a video-based search tool offers direct benefits for the Flemish DHH community as it would make the search functionality in the VGT dictionary fully bidirectional. As the text-to-sign search function is currently more accurate, the dictionary is mostly used to look up the corresponding signs to a Dutch word. The fact that this feature is mostly useful for non-signers and novice signers is reflected in the user base, as most users of the dictionary are hearing VGT learners. Enabling reliable sign-to-text search plays a big part in expanding the user base with more DHH users, especially those with lower proficiency in Dutch, as it would allow them to use the dictionary in their native language. However, broader adoption among DHH users also depends on complementary developments: expanding the dictionary with more specialised vocabulary, and gradually introducing explanatory elements such as definitions and example sentences in VGT. Together, these improvements would allow the dictionary to better serve both hearing learners and native VGT signers. In addition, the technical progress that the sign-to-text search will also enable a range of other usage scenarios, such as lexical research, e.g., to search for similarities among dictionary signs, or the development of VGT learner support tools, driven by the Flemish DHH community, and VGT teachers.

Feasability: the work of De Coster and Dambre [36] employs a version of the VGT–Dutch dictionary collected (with permission) in 2023, containing 10,235 unique glosses, each with a single example video and its most common meanings in written Dutch. At the time of writing, the dictionary already contains 11,248 signs: 1013 new signs have been recorded and published since then.[6] Each entry is also annotated with additional meta-information, such as its most discriminating sign parameters (e.g., hand shape(s) or location), which can be used to perform dictionary queries. The process of standardisation of VGT is still in progress. The dictionary is a collective effort of many members of the Flemish signing community, coordinated by VGTC. Collection and verification efforts lead to the regular addition of new signs. Under these conditions, a practically useful sign-to-text search tool must therefore be *truly scalable*, in the sense that it should enable the seamless addition of new signs without (frequent) retraining.

From a data perspective, supervised learning or even few-shot learning are not possible for this use case: for more than 90% of the dictionary entries, the example in the dictionary is the only one available. The Euclidean distance-based

[6] This illustrates the need for a *scalable* ISLR solution that is easily adapted to the recognition of new signs. SignBuddy is such a solution, and we will update to the complete dictionary when the application is finalised—and regularly update thereafter.

vector search approach used by De Coster and Dambre [36] matches the restrictions imposed by the use case and was therefore selected as the basis for our solution. However, considerable improvements are necessary. First, their training dataset was not diverse enough to represent the full spectrum of sign parameters used in the dictionary. Second, a more diverse dataset may require a more powerful model to capture those additional richness of the data.

Requirements: a requirement for adoption of the developed search functionality is that the accuracy of our solution must be sufficiently high. Both parties agreed that the search tool should present the top-k most similar signs in the dictionary, and VGTC defined the quality criteria the search function should achieve in order to be integrated into the dictionary:

- the top-k predictions should fit on a single screen, in the layout template of the online dictionary (this restricts k to 6 in a relaxed layout and 9 in a tight layout);
- the top-k recall is ideally above 70% when searching in the entire dictionary, with an absolute minimum of 2 out of 3 (66.6%), for well-executed signs.

3.3 The need for crowd-sourced data collection

The minimum requirements, corresponding to a minimum Recall@k of 66.6%, puts forward the need for test data that consists of dictionary queries of signs that were executed correctly (i.e., by people with a certain level of sign language proficiency) for a wide range of dictionary signs and that contains variability that is representative of the envisaged usage.

The evaluation set used by De Coster and Dambre [36] only covered ten unique signs, executed by non-signers (so possibly not always accurately executed). It therefore does not allow us to monitor our targeted Recall@k. As a step towards achieving our primary goal, high-quality VGT evaluation data was needed, with recording conditions that were similar to those of future dictionary searches (i.e., data collected *in the wild*).

Finally, the credibility of technological solutions can be greatly enhanced if the mistakes made by the model make sense to the end users. Technically, this means that, if a human would perceive two signs to be similar, then the model should also consider them similar (and vice versa). Again, in order to assess whether this is the case, human evaluation data is crucial.

Based on the analyses above, VGTC and UGent decided to develop SignBuddy, a data collection environment that encourages signing users (DHH-C), to contribute their sign language expertise for the development and iterative improvement of the envisaged dictionary search functionality. Since the VGT dictionary is not only used by signers but also by people with little to no pre-existing level of sign language proficiency, we extend our data collection to both groups in order to be able to monitor the sensitivity to deviations in signing quality. To distinguish between these groups, participants are asked to indicate their sign language proficiency (further referred to as *sign level*).

To collect labeled data, users are shown the video of a sign from the dictionary and asked to record themselves while executing that sign. SignBuddy also leverages the VGT-expertise of DHH-C and members of H-P with signing experience to assess whether the model's mistakes make sense. In particular, we ask the user four questions about the model's most highly ranked mistake:

- Was the hand shape of one or both hands correct?
- Was the signing location of one or both hands correct?
- Was the movement of one or both hands correct?
- Was the mouth pattern of the sign correct?

These questions allow us to assess which sign language parameters[7] are especially problematic for the deep neural networks that perform ISLR. Such human evaluation data is crucial for making informed decisions with regard to the used machine learning models.

To realise SignBuddy, VGTC and Ugent jointly (and successfully) applied for funding for AMAI "Wat Gebaar jij?!" [50], funded by Flemish government with VGTC as a coordinator. Both partners conceived and defined the project outline and planning together. Besides the development of the SignBuddy data collection tool, the project also included the data collection and analysis itself, and the use of this data to optimally adapt the pre-existing technology of UGent for incorporation into the VGT dictionary. This paper reports on the process and results of this project.

4 The realisation of SignBuddy

This section covers all the practical aspects of SignBuddy's development. First, we list the design principles that steered this development phase in Sect. 4.1. These principles bridge the previous section on the ideation of SignBuddy (Sect. 3) into this section, and describe how the requirements are transformed into actionable concepts. Subsequently, in Sect. 4.2 we disclose the technical aspects about searching through vast vocabularies, and follow with an overview of the interface in Sect. 4.3.

[7] Signs are often described by five articulatory parameters: handshape, orientation, location, movement, and non-manual signals [48, 49].

4.1 Design principles of the SignBuddy interface

As emphasised in the previous section, SignBuddy is a tool specifically developed with and for the DHH community. Its interface is hosted at VGTC, as part of the VGT-dictionary website. Given its intended user base, great care has been taken to ensure that both the functionality and the design of SignBuddy's interface align with the community's preferences and requirements. We highlight the following seven design choices that benefit the user experience and simultaneously maximise the informativeness of the collected data. While these decisions were made jointly, VGTC was in the lead for this part.

1. *Accessibility* The tool's visual design and interactive experience have been crafted to mirror those of the existing VGT-Dutch dictionary website. This consistency makes navigation more intuitive for users who are already accustomed to the dictionary's interface. The tool is publicly available and does not require user credentials, which removes potential barriers to entry and ensures equal access for all users.
2. *Inclusivity* The commitment to inclusivity is reinforced by the bilingual presentation of content in both VGT and Dutch, accommodating individuals with diverse linguistic backgrounds. This is important, because reading acquisition can present significant challenges for DHH individuals [51]. Dutch and VGT enjoy an equal prominence in SignBuddy. Thereby, the tool supports accessible information and a user-friendly experience for the DHH community, as well as hearing persons.
3. *Usability and clarity* Every interactive element within the interface is paired with an intuitive icon that visually represents the corresponding action. This approach streamlines user interactions, making the tool more accessible to a broad range of users. Some examples are listed in Fig. 1. Beyond visual design, all information is provided in both VGT and Dutch, ensuring clarity and inclusiveness. Transparency is a core principle of co-creation and a prerequisite for informed participation. SignBuddy therefore communicates clearly how user data are processed and anonymised before being used in AI-based sign language tools. This openness not only builds user trust but also strengthens AI literacy by helping participants understand how machine learning models function and evolve. In this way, users become informed collaborators who both contribute to and learn from the technology they help shape.
4. *Privacy* We prioritise user privacy at every stage of development. SignBuddy does not process or store raw video data, but collects only anonymised key-point sequences that are extracted on the browser, on the user's computer. Alongside each sequence is stored its sign level, label (i.e. gloss), top-6 predictions with probabilities, timestamp, user feedback, and a unique ID. This ID allows for accurate data management and processing, but cannot be linked back to user sessions. Additionally, feedback and sign-level selection are provided exclusively through checkboxes. This eliminates the need for textual input and minimises the risk of personal identification. Moreover, since reading and/or writing text can be a barrier to some DHH persons [51], such checkboxes with associated icons make the application more inclusive.
5. *User engagement* A user is more likely to continue recording signs when the system is correct most of the time. As the data collection through SignBuddy is a step in the development of a robust dictionary search tool, we boost perceived performance (as opposed to actual performance) in a number of ways. First, instead of only showing the best match found by our model, we present the top-6 predictions. To strengthen the perception of "success" and increase the gratification of recording signs, we added a large green checkmark to the correct sign if it is present in this top-6. The restriction of the number of signs that are shown to 6 was dictated by considerations of accessibility and visual attractiveness. More specifically, we ensured that all presented signs fit on a single screen (without the need for scrolling), to improve user satisfaction like mentioned by Hassan et al. [52]. Between the two feasible options in the layout template used for the VGT-Dutch online dictionary (top-6 and top-9) we opted for the one with the largest videos. However, as discussed in Sect. 4.2, the initial model used in SignBuddy is not yet good enough to achieve a gratifying (top-6) success rate. This is not surprising, as the purpose of collecting data through SignBuddy is to improve the model's performance with the gathered insights from user data. In order to emulate a

Fig. 1 Composite of interactive elements in SignBuddy, illustrating the informative pictograms

boosted model performance, SignBuddy's backend performs dictionary search not in the entire dictionary, but in a large subset of the dictionary. By adapting the size of this dictionary subset, we create a positive feeling for SL technology, avoid frustration and increase the likelihood that users will be willing to contribute more than one sign.

6. *Variety of collected data* We want to cover as many signs as possible. However, if we immediately cover all signs, the number of collected samples per sign will be low. For this reason, we select signs from a limited set of 200 signs at any given time. Once ten samples of a specific sign have been collected, it is replaced with a new sign from the dictionary. This dynamic rotation ensures a diverse range of signs is sampled with a representative number of samples for each entry. At present, newly added signs are randomly selected. In the future, we may slightly steer this selection process to avoid selecting signs that are very similar to the ones we already have and expand towards types of signs that are not yet represented in the collected data.
7. *Pertinence of collected data* Aside from the sign recording, SignBuddy also collects other user inputs, i.e., whether the user is proficient in sign language and user feedback on sign similarity. We ensure that the collected data is workable by constraining the number of input possibilities using checkboxes. Dictating the shape by which this information is provided, makes the collection more interpretable. Additionally, by questioning the highest-ranked erroneous predictions, we ensure that the collected feedback reflects relevant model confusions. This brings relevant shortcomings to light, driving SLP research in a direction substantiated by user feedback.

4.2 Searching through the VGT dictionary

To provide an initial version of the search functionality that underlies SignBuddy, we have built upon the one-shot classification approach described in [36]. This consists of two steps: *initialisation* and *inference*. In the initialisation step, a pretrained model (see the following paragraph) converts all dictionary videos (later also called the *key*s) into embeddings using a pretrained neural network. During the inference step—depicted in the right section of Fig. 2—the same embedding model converts the current input video, *the query*, into its corresponding embedding. Finally, the dictionary entries that are most similar to the query are identified by comparing the query embedding with all key embeddings. In contrast to [36], which compared embeddings with the Euclidean distance measure, this comparison uses the attention mechanism described by Bahdanau et al. [53] as this resulted in a slightly better performance.

A robust sign embedder is the first crucial element of this technique. We used a similar architecture as in the original approach [36], but achieved a considerable improvement in performance by selecting a more suitable dataset for pretraining. The original model was trained on a VGT Corpus [37] dataset of cut-out segments from continuous signing. It contains samples from 292 signs, but its class distribution is very unbalanced and the most common signs are also the simplest ones: various forms of pointing signs. We conjectured that the resulting lack of richness could be a core reason for the rather poor performance of the resulting search model. An additional cause may be the fact that the cut-out segments retain co-articulation transients from preceding and succeeding signs and are not recordings of signs in their citation form. This is a form of variability that is not present in dictionary videos and therefore reduces the classification accuracy through domain mismatch.

While a much larger VGT dataset could solve these issues, this is not available. Instead, we released the language restriction and opted instead for a rich dataset with many variations between different signs. For this reason, we chose ASL-Citizen [54] due to its sizeable vocabulary and balanced class distribution. This dataset consists of isolated sign recordings. Additionally, ASL-Citizen is a crowdsourced dataset described as a collection of "signs in the wild", where DHH individuals record themselves signing via webcam, ensuring a diverse and representative sample.

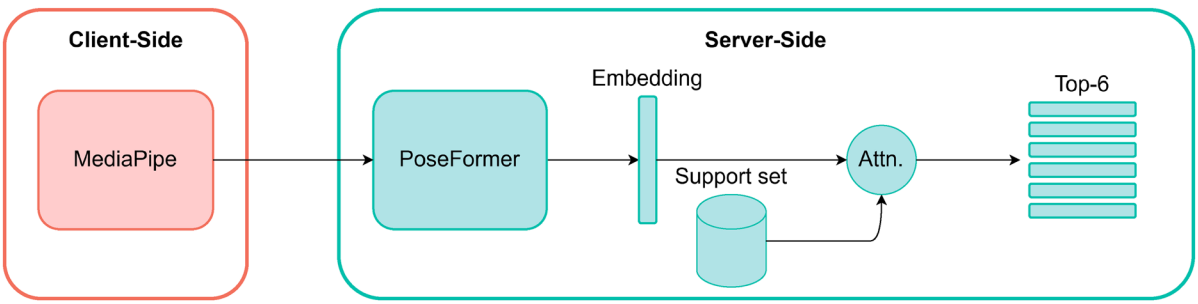


Fig. 2 Internal processing pipeline of SignBuddy: Keypoint extraction is performed locally on the user's device, and the recognition and prediction are conducted on a remote server

Both of these properties make it better aligned with our evaluation setup than the original VGT corpus dataset.

UGent trained and optimised an embedder using ASL-Citizen classification, on which it achieves state-of-the-art performance. More details about this can be found in Appendices A and B. To evaluate the quality of the new pretrained model as an embedder for dictionary search, we used the same lab-sourced validation set as in [36]. Figure 3 compares its performance to that of the original model from [36]. The results show a huge improvement, confirming our assumption that the size of and variation in the pretraining dataset and its alignment with the evaluation task are more important than using a dataset from the same sign language. These results also demonstrate the large impact on perceived success rate by displaying the 6 most similar signs from the dictionary, instead of only the most similar one.

As is also clear in Fig. 3, the success rate of the queries drops considerably as larger parts of the dictionary are searched. Based on our small validation set, the Recall@6 on the full dictionary for PF-ASL is estimated at 0.696. However, this validation set was originally created to perform user experiments among non-signers, so the selection of signs favoured signs that were easily recognizable and not very complicated or nuanced. We therefore suspected that the difficulty of this set is not representative for the whole dictionary, a suspicion that is confirmed by the initial results described in Sect. 5. By initially constraining the search space to a subset of one thousand signs (from 11,248), UGent and VGTC agreed to build in some margin to guarantee a sufficiently high perceived success rate when SignBuddy was released. For a vocabulary of 1000, this yields a Recall@6 of 0.923 on the validation set (compared to 0.696 for the complete dictionary). As we have now established that our model has achieved the required minimal Recall for well-executed signs (cf. Sect. 6), we will now remove this restriction.

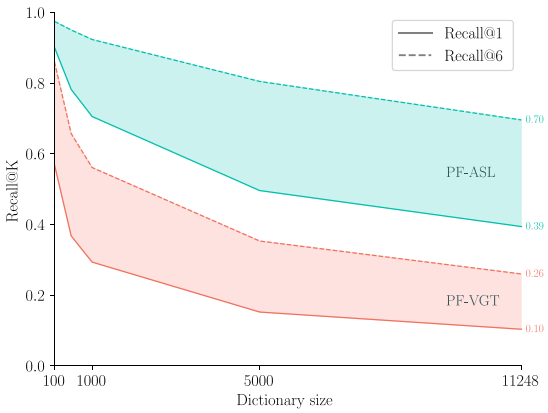


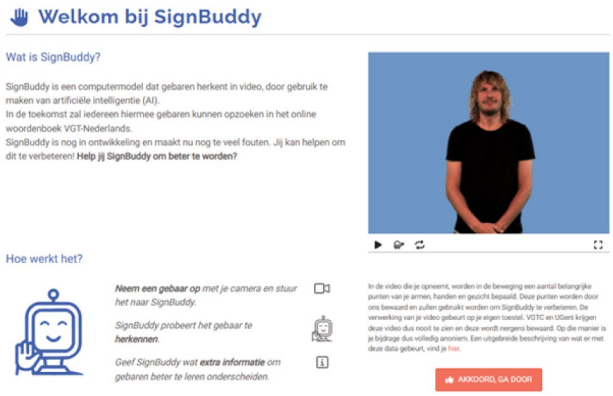
Fig. 3 Recall@1 and Recall@6 for dictionary search on the lab-sourced evaluation set. PF-VGT refers to results with the original embedder from [36], whereas PF-ASL relates to results obtained with the new embedder, trained on ASL-Citizen

4.3 Overview of the SignBuddy interface

The dictionary search approach detailed above is presented to the user with a web interface integrated into the existing VGT-Dutch dictionary. During the integration phase, members of DHH-C were consulted to give feedback on the user friendliness and overall appearance of the interface. Figure 4 outlines the user's interaction with SignBuddy, which consists of six sequential steps. All information is provided in written Dutch and in signed VGT, in some cases clarified with pictograms. First, the user receives usage information and acknowledges participation (Fig. 4a). The user is informed about SignBuddy's purpose, and about the data privacy, i.e., that the recorded videos are only processed on their own device but their pose keypoints are processed and stored on a remote server. Users can access more information by clicking the highlighted *"hier"* (here), which directs them to a second page containing detailed explanations-available in both VGT and Dutch-about the project, its goals, and how their data is handled. Upon providing consent by clicking the acknowledgement button, the user proceeds to the next screen.

On the second screen (Fig. 4b), the user is asked to indicate whether they are proficient in sign language by selecting one of three options: *'No'*, *'A bit'*, or *'Yes'*. We deliberately refrain from asking about users' hearing status, as this does not necessarily reflect their proficiency in VGT. For instance, individuals who suffered from hearing loss later in life may not have acquired any VGT. Therefore, the self-reported sign level is far more informative for this study. While the ultimate goal of SignBuddy is to enable accurate dictionary search, the current phase focuses on collecting targeted signs for analysis. For this purpose, the second screen also presents the prompt, i.e., the sign the user is asked to replicate. Admittedly, providing an example sign limits variability of the collected signs. However, it also ensures a more balanced sampling of signs across participants. Moreover, this inter-sign limitation does not hamper the intra-sign variation—i.e. variation of execution. Individual signing styles remain distinct, which is equally important to capture when collecting representative sign data.

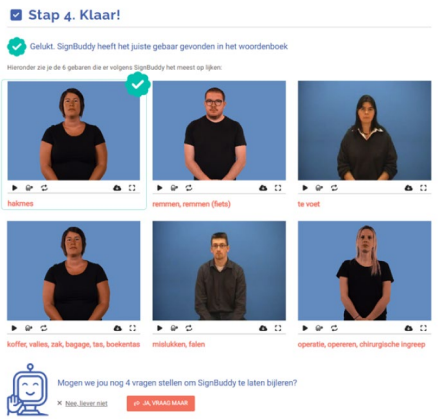
On the next screen (Fig. 4c), the system allows the user to record themselves performing the given sign. The example video is still available on this screen, in case the user wants to review it. When the recording button is pressed, a three-second countdown begins, followed by four seconds of recording time. During this window, the user sees themselves with keypoints projected onto their body. They can redo the recording as many times as needed before pressing the send button. Only the final attempt is transmitted to the remote server.



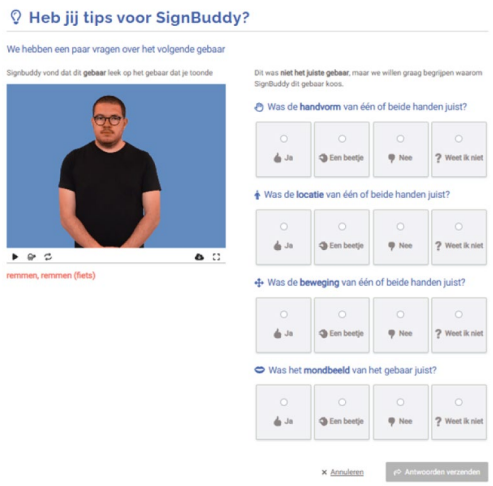
(a) The user is provided with usage information and acknowledges participation.

(b) The user is prompted to indicate their level of sign language proficiency (*'None'*, *'Some'*, *'Fluent'*), while the specific sign they need to perform is displayed.

(c) The system allows the user to record themselves performing the sign (and view the example production again as many times as they want).



(d) SignBuddy generates predictions for the top six matches and allows the user to complete a questionnaire.



(e) The user optionally provides feedback about dictionary lookup results.

(f) The user is thanked for their contribution and redirected to the first screen.

Fig. 4 Overview of the SignBuddy interface

The web interface visualises predictions for the top six potential matches on the fourth screen (Fig. 4d). The correct sign (if present) is indicated with a large green check mark. The user is also asked to give feedback on the predictions by pressing the orange button at the bottom of the screen.

By agreeing, they proceed to the fifth screen (Fig. 4e), where they are asked to answer some questions about the highest-ranked erroneous prediction generated by the system. The user is presented with the video for this sign and asked to evaluate how accurately the hand shape, location, movement, and mouth pattern match the sign they performed. For each feature, users can select from four options: *'Yes'*, *'Slightly'*, *'No'*, and *'I don't know'*, each accompanied by a corresponding icon. After giving feedback or indicating they do not want to give feedback, the user is thanked for their contribution and asked if they want to progress to another sign (Fig. 4f).

The technical steps that are executed between screens three and four are illustrated in Fig. 2. The user's keypoints are predicted locally on their device to ensure anonymity and reduce server-side processing. These keypoints are then transmitted to the remote server and processed using a PoseFormer model to generate embeddings for one-shot classification, as detailed in Sect. 4.2. Finally, the system returns the top six ranked matches to the user. The keypoints, the sign that was executed and the feedback are stored on the remote server. This allows us to use the recorded data to evaluate future improved models and compare model alignments using the user feedback.

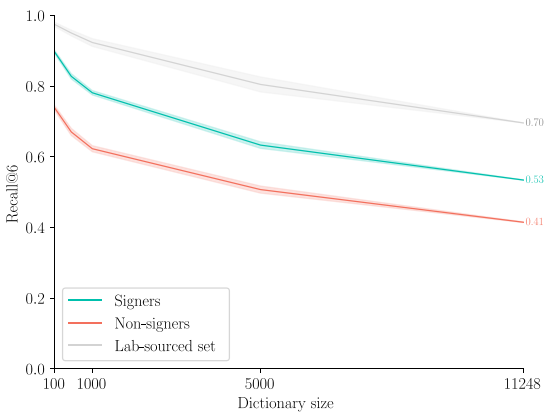


Fig. 5 Mean Recall@6 with (±2 standard deviation bands) for large dictionary search with set permutation of the SignBuddy-collected data, based on the level of sign language proficiency. We repeat the mean Recall@6 curve from Fig. 3 as "lab-sourced set"

5 What we can learn from the collected user data

The following section employs the same structure for each subsection. We first analyse the results and then summarise all insights briefly. In Sect. 5.1, we report the number of collected signs and present a brief evaluation of the model to offer an overall perspective. Thereafter, we give a more thorough evaluation of the employed model (Sect. 5.2). Next, we qualitatively analyse the collected keypoint data (Sect. 5.3) and conclude this Sect. by evaluating the user feedback (Sect. 5.4).

5.1 Data properties and model performance

5.1.1 Analysis

At the time of writing, approximately one year after SignBuddy's public release, a total of 1868 samples have been collected, with examples of 377 unique signs in total.[8] 1120 of those samples come from users who indicated they have "proficient in sign language",[9] (318 unique signs) and 748 from people without sign proficiency (327 unique signs). The intersection of both sets contains 268 unique signs. On average, there are 5 samples per sign class.

The samples provided by people with sign language proficiency were mainly collected during a few organised collection moments for DHH users, but a call to contribute was also distributed through social media. Similarly, a large fraction of the samples from non-signing hearing persons were collected at events for science popularisation, in addition to calls through various channels. Finally, a link to SignBuddy was added to the starting page of the VGT dictionary, to collect samples from people who already know and use it.

Figure 5 shows the average dictionary search performance for each subgroup (henceforth called *signers* and *non-signers*), and compares it to the search performance on our validation set. Each sample is tested against 100 independently sampled subsets of the dictionary. Over these separate tests, we report the mean Recall@6 along with its standard deviation.[10] Furthermore, for the samples collected

[8] As SignBuddy is still active, the data set is still growing. The numbers reported in this section will be updated if/when the paper is accepted. We also plan to publicly release a considerable part of the collected data for research purposes.

[9] Participants could indicate their proficiency level of sign language by selecting one of three options: *Yes A bit*, or *No*. Since the response *a bit* was considered too ambiguous, these participants were grouped with those reporting no prior sign language proficiency, effectively ensuring the proficiency level of the group with sign language proficiency.

[10] The largest dictionary size, comprising the entire dictionary, cannot be randomly sampled. Therefore, its standard deviation is reported

from signers, our approach achieved a Recall@6 of 0.534 on the complete dictionary, compared to 0.414 for non-signers. First, these results are a lot lower than what we obtained for the validation set (also shown in Fig. 3), which confirms our initial suspicion that the validation set was relatively easy, in comparison with a randomly sampled subset from the entire dictionary.

Also, we observe a significant difference between the success rates for signers and for non-signers. This difference may in part be explained by the different settings in which the (bulk of the) data for each subgroup has been collected and possibly also by different motivations of the participants in each group. However, as we were present to observe the recording of a large part of the collected samples, we have noticed that at least part of the difference is due to the fact that non-signers often miss the finer nuances of correct sign execution.

5.1.2 Insights

It is crucial to collect a varied and representative evaluation set: the validation set of [36] was too narrow and provided an overly optimistic view of model performance. Moreover, our decision to ask users for their signing proficiency is supported by our data: there is a clear difference between the model's performance for signers and non-signers. The collected signer proficiency metadata facilitates our error analysis.

5.2 Analysing the quality of the embedder

5.2.1 Analysis

One of the things we may learn from the SignBuddy data is the suitability of the used training set. In particular, we expect that enlarging the training set, e.g., by combining data across multiple sign languages, would further improve performance. However, earlier attempts at doing so yielded disappointing results. This suggests that we need a more targeted approach of adding data for specific signing patterns for which the initial SignBuddy model does not perform well.

A first step towards this is to analyse how well the sign patterns that occur in the dictionary align with those in the training set. When training a model on a given classification dataset, it is optimised to discriminate between the classes in that dataset and may yield an unreliable view on similarity for signing patterns that are very different from the signs in the training set.

Based on these considerations, we would expect the model to not work as well for VGT dictionary signs that are very different from signs in the training set and work *best* for signs that correspond to *dense regions in the training set*, i.e., regions in which multiple similar signs occur (because the model will be proficient at recognising such signs). In addition, since our approach searches the dictionary for the signs that are most similar to a query, we expect it to be *less effective* in discriminating between dictionary signs that are very similar, which correspond to *dense regions of the dictionary*.

In order to quantify whether these insights are reflected in the collected data, we used a measure for the *local density* of a sign data set around a given embedding vector. The measure we use is based on *cosine similarity*,[11] which is one of the most commonly used similarity measures to compare vectors. More specifically, we identify the six data set samples that are most similar to that vector and use the similarity to the sixth nearest neighbour as a local density measure. We specifically use the sixth nearest neighbour to align with SignBuddy's interface, which prioritises the top six predictions, thereby reflecting interactions between the users and SignBuddy. To quantify the local density of an embedding in the dictionary, we compare with the key embeddings. To evaluate the local density in the training set, we compare with class embeddings, for which we use the average embedding across all training set samples of each class.

We first analyse the overall alignment between the local densities of the training set, the dictionary and the signs that occur as queries in the data collected thus far in the two panels of Fig. 6. The coordinates of each point in these plots are the local densities of the key embedding vectors in the dictionary (on the x-axes of both panels) and in the ASL-Citizen training set (on the y-axes). Lower values of these densities correspond to sparser regions of the embedding space, while high values correspond to dense regions.

Both panels show the collected data as coloured dots, each with a different set plotted in gray in the background. The left panel shows the local densities for all dictionary keys, while the gray dots in the right panel correspond to the same densities for the training set classes. Each coloured dot corresponds to a sign that occurs in the collected data. Its colour reflects the average cosine similarity of the query embeddings for that sign and its corresponding key embedding. It is a measure for how well our embedder perceives the queries and the key for that sign as similar.

as zero. In general, the standard deviation remains relatively small. For all three test sets, the largest standard deviation occurs at a dictionary size of 5000. Specifically, we observe standard deviations of 0.0046 and 0.0050 for the signers and non-signers, respectively, and 0.0106 for the lab-sourced set. This observation is expected, since the SignBuddy-collected data covers a wider range of unique signs, and therefore is less sensitive to the dictionary composition.

[11] Cosine similarity is defined as the normalised dot product between two vectors. Its values lie between -1 (opposite vectors) and +1 (similar vectors). Formula: $\cos(\mathrm{A}, \mathrm{B}) = \frac{\mathrm{A} \cdot \mathrm{B}}{\|\mathrm{A}\| \|\mathrm{B}\|}$

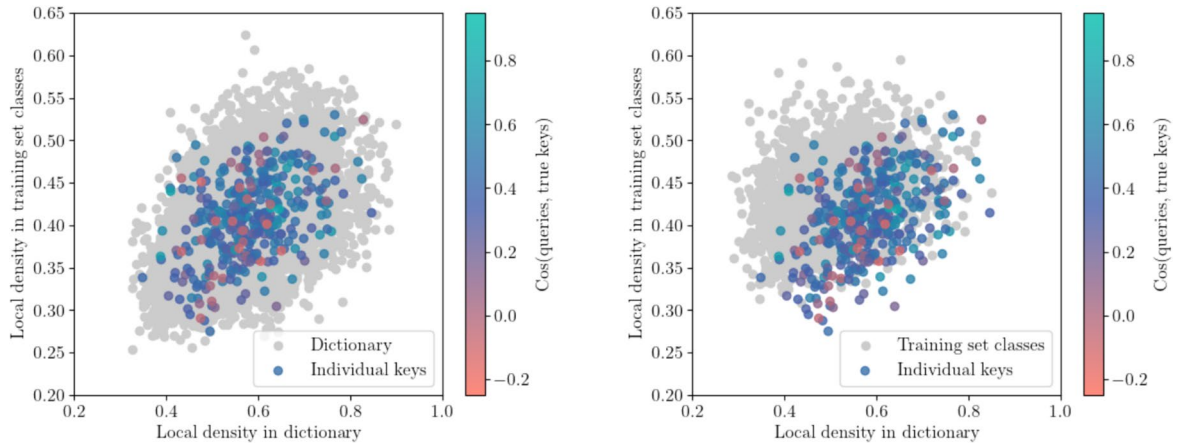


Fig. 6 Visualization of local density in the dictionary and the training set. (Left) Alignment of dictionary queries (coloured) with the dictionary keys (grey). (Right) Alignment of dictionary queries (coloured) with the training set classes (grey). The colours indicate the mean of all cosine similarities of one key to all of its queries

In the left panel, each dot (coloured or gray) corresponds to a sign example from the dictionary, so in this case, the coloured dots are a subset of the gray dots in the background. It illustrates that the collected classes span a diverse subset of the dictionary, exhibiting no apparent outliers while encompassing varying density levels of the dictionary entries—from tightly clustered to more dispersed regions. In contrast, when compared to the same measurements for ASL-citizen (grey points on the right panel), a considerable part of the collected data originates from sparser regions of the training set embedding space. This aligns with expectations, as ASL and VGT are distinct sign languages with divergent linguistic structures, which likely reduces embedding similarity when a model trained on one language (e.g., ASL in PF-ASL) processes signs from another.

However, what Fig. 6 does not show, is any clear relation between local densities and the average similarity between queries and keys (i.e., there is no distinguishable pattern in the colours of the points). Based on visual inspection of the keypoint sequences that are used as model inputs, we have noticed that there is often considerable difference between different queries for the same sign. The results in Fig. 6 suggest that, based on the current sample, these individual differences are more important for the overall performance of the model than the general misalignment between the training set and the dictionary. Some of these differences and their underlying causes will be highlighted in Sect. 5.3.

To get a better understanding of the relation between the recorded signing patterns, the alignment between training set and dictionary and the recognition quality, we now analyse all individual collected samples (so without averaging them per sign). In addition to the similarity between query and key (again used as marker colour), we now visualise the *rank*[12] of the recognition to quantify the recognition quality for each query in Fig. 7. Each of the three panels in Fig. 7 gives a different view on regions of poorer model performance. Since we use Recall@6 as a global quality measure, all predictions with ranks up to 6 count as *successful*. In all panels, as expected, we see a clear correlation between the similarity between query and key and the recognition rank, but together, they give a more nuanced view on model performance. In order to establish whether a trend exists between local densities and model performance, the median rank for different regions of the local density is shown in black on each panel.

In the top panel, the x-coordinate of each point reflects the local density *in the dictionary around the key* embedding for that sample. Based on the reflection that in dictionary regions with higher density (multiple similar signs), classification would be more difficult, we would expect to see a median trendline that rises towards larger densities. Such a trend would mean that our model needs to be better adapted to the specific signs in the VGT dictionary, for example by language-specific fine-tuning. In practice, the increase we see for the current sample is very small and probably not significant. We conclude that language specific fine-tuning is not the first priority on the road to model improvement.

In the middle panel, the x-coordinate of each point reflects the local density *in the training set around the key* embedding for that sample. From Fig. 6, we learned that the dictionary and the training set do not quite align: part of the dictionary signs are in sparse regions of the training set. Our embedder may be less accurate in those regions: we would expect to see a median trendline that decreases towards higher densities in the training set. We indeed observe such

[12] The rank of a search result is where it appears in the list of all search results: if the result is first, its rank is 1. If it is second, its rank is 2, and so on.

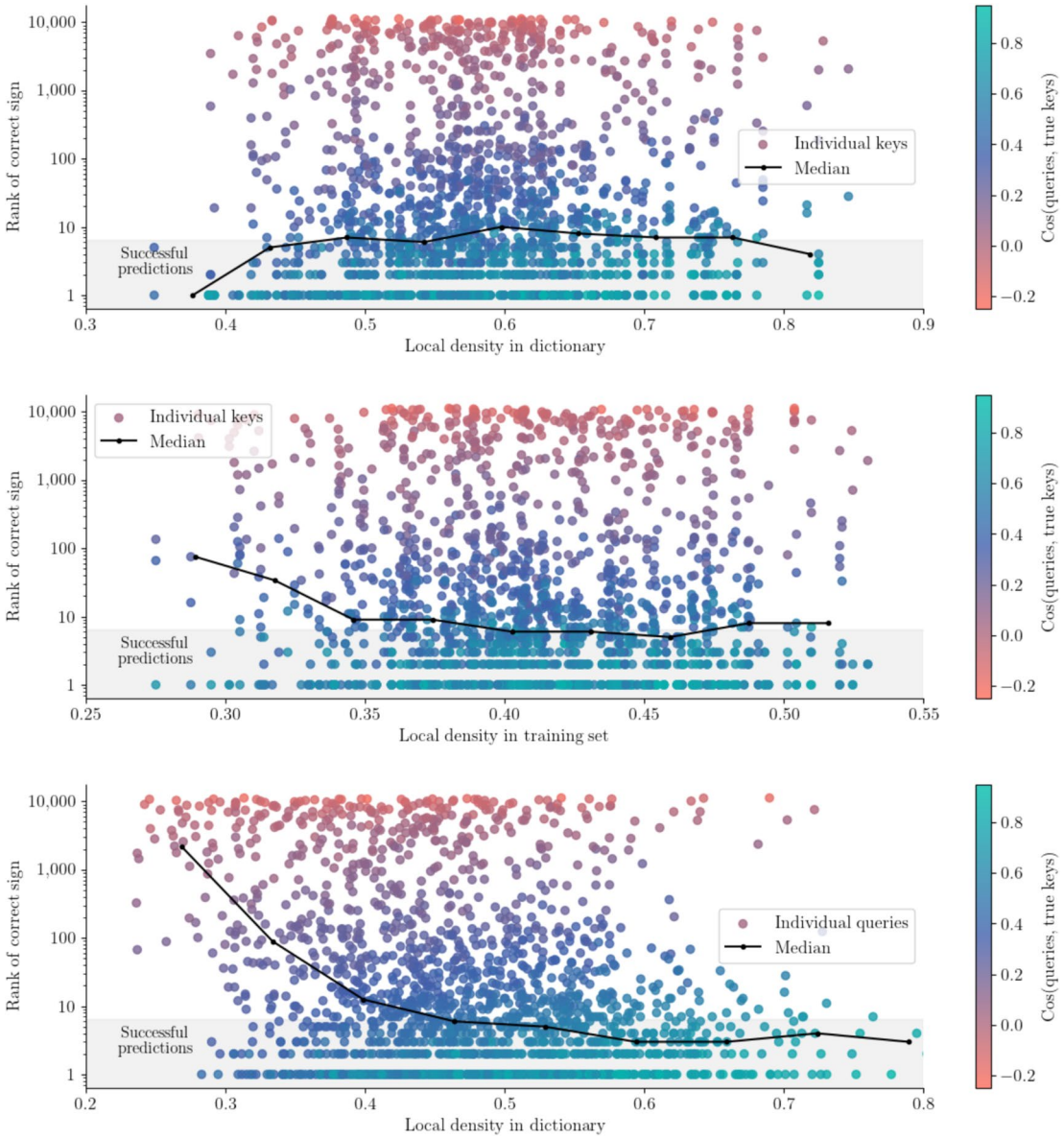


Fig. 7 Rank as a function of (Top) local density in the *dictionary* around the *key* embedding, (Middle) local density in the *training set* around the *the key* embedding, and (Bottom) local density in the *dictionary* around the *query* embedding for each collected sample. This figure illustrates how the density in the dictionary and training set influence the search results and where the correct search result appears

a trend, and also see an increased rank for very high density regions (the trendline goes up to the right), for which we do not have an immediate explanation. However, we still conclude that it may be useful to try to extend the training set in a targeted way with classes from other data sets (possibly in other languages) to extend the variation between signs.

Finally, in the bottom panel, the x-coordinate of each point reflects the local density *in the dictionary around the query* embedding for that sample. While in the previous two panels, we see no clear relation between colour and local density, we see on this panel that executions that poorly resemble their dictionary example (high rank) tend to cluster to the left (low density). Similarly, executions with high similarity (low rank) to their corresponding key occur more frequently in high density regions of the dictionary. The median trendline confirms that points with very low values typically result in poor recognition (high rank), while the relation between rank and density disappears for higher density regions. From Fig. 6, we already learned that some keys themselves lie in low-density regions. However, this plot shows a very strong impact on performance of samples that are more misaligned with the dictionary than their corresponding key. Surprisingly, we also see a few samples with very poor recognition in high density regions. Overall,

we conclude that properties of (the keypoints of) individual samples have the strongest impact on model performance. These will be further investigated in Sect. 5.3.

5.2.2 Insights

The pretraining of our model on ASL and using it on VGT leads to the VGT dictionary entries occupying a sparser region of the model's embedding space due to the languages' differences. Yet, the majority of the errors made during retrieval stems from (lack of) query quality, and not from this language mismatch. We do not observe a trend that would suggest that we would benefit from fine-tuning our model to VGT data. However, targeted data collection to increase the variety of the pretraining set by adding new sign categories could help improve performance.

5.3 Qualitative assessment of failure cases

5.3.1 Analysis

As the previous analyses indicated that specific properties of individual keypoint recordings have a strong impact on model performance, we performed a visual inspection of the collected keypoint sequences, in comparison with those of their respective dictionary entries. Like the section about the driving design choices (Sect. 4.1), we structure this section around shared concepts between the observations. We provide example illustrations of failure cases due to some of the listed effects in Appendix C.

1. *Keypoint quality* has a large impact on model accuracy. Unlike in lab-controlled setups, SignBuddy's real-world deployment introduces considerable variability in recording conditions, such as lighting conditions, camera type, or distance from the camera. When these do not match the usage conditions of MediaPipe, they often result in incomplete or noisy hand landmark detection. This poses a significant issue, given the reliance of sign language recognition on precise hand articulation. If hand keypoint inaccuracies occur in most or all of the frames that contain the most essential parts of the sign execution the sign becomes unrecognisable from the keypoints, even for humans. We found several cases in which frequent failures to detect hand keypoints in key frames directly degraded the recognition accuracy. In some cases, other systematic errors in pose estimation (e.g., misaligned joints or spurious keypoints) further compound this challenge, particularly for signs in the VGT dictionary that require fine-grained manual distinctions. A second major concern is frame inconsistency. MediaPipe's in-browser real-time processing often fails to keep pace with incoming video streams, leading to dropped frames and unstable effective frame rates. The top part of Fig. 8 highlights this issue, demonstrating a misalignment between the nominal frame rates of common webcams and the actual number of frames in the collected keypoint sequences which all originated from a recording of the same duration. This inconsistency disrupts the temporal convolutions of PF-ASL, which assume uniformly sampled input.

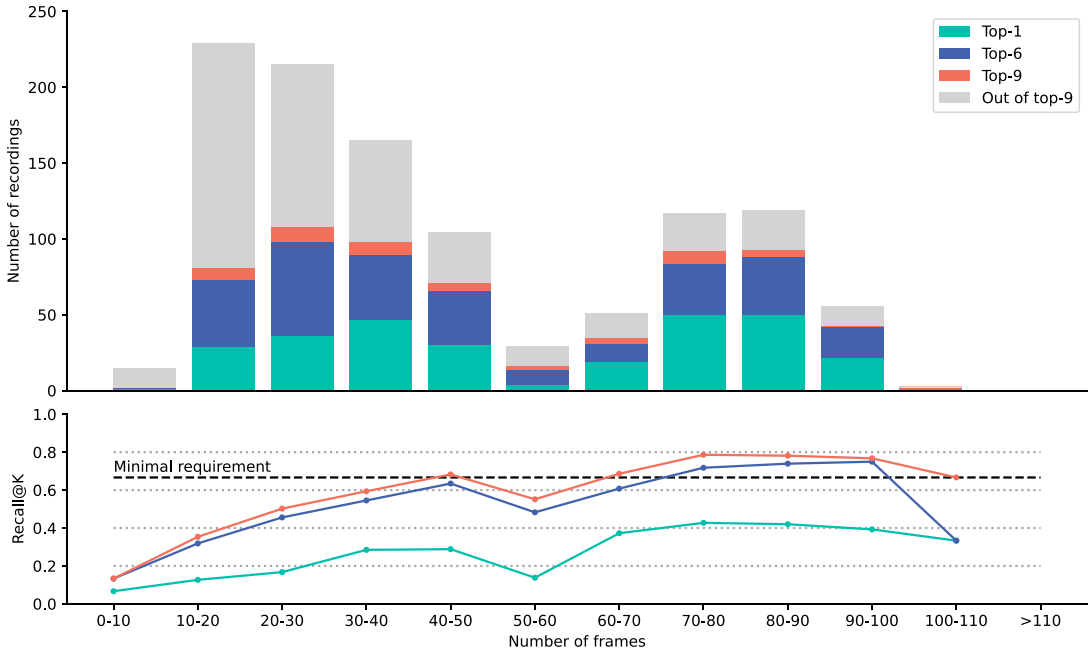


Fig. 8 The top plot shows the number of collected samples of signers for each frame count. Colors indicate the proportion of correctly predicted samples within a given top-K rank. The bottom plot shares the same x-axis and presents the corresponding Recall@K values

Compounded by the inherent limitations of consumer-grade webcams (typically 15–30 FPS), the resulting recorded keypoint sequences often contain insufficient temporal resolution to capture rapid sign movements. The bottom part of Fig. 8 further reveals a positive correlation between the number of frames and Recall@K across different K values, underscoring the need for a more robust keypoint recording pipeline.

2. *Temporal instability and tracking artefacts* represent a second category of challenges. These issues, while affecting a smaller subset of the data, manifest abruptly and degrade prediction reliability. First, spurious keypoint trajectories occur when hand landmarks exhibit sudden, unnatural shifts-for example, rapid movements toward the centre of the frame before, during, or after sign execution. These anomalies likely arise from pose estimation errors during rapid motion or partial occlusion. Second, hand identity swaps—where left and right hand keypoints are misassigned in individual frames—disrupt the continuity of manual feature tracking. Such swaps are common in signs involving crossed arms or overlapping hands, where pose and hand detection struggle to disambiguate limbs. A final artefact in this category is boundary-induced jitter, where keypoints fluctuate erratically as hands enter or exit the camera's field of view. This occurs because pose estimators like MediaPipe extrapolate keypoints for partially visible limbs, without regarding temporal soundness. Collectively, these artefacts corrupt the spatiotemporal coherence of input signs, challenging PF-ASL's ability to isolate discriminative kinematic patterns.
3. *Variation in sign execution* represents a third challenge for recognition systems. Here, we distinguish two cases. In the first case, signs are executed within acceptable human perceptual tolerances of the example video in the dictionary, but small rotations of a hand or non-frontal camera angles (e.g., a signer oriented obliquely relative to the camera) can alter the 2D keypoint projections[13] used by PF-ASL, leading to erroneous interpretations. Figures 12 and 14 show examples. These errors highlight the model's sensitivity to camera perspective and underscore the need for more robust 3D keypoint estimators. A second case is related to the allowed variability in some sign parameters. While for some signs, these are strongly specified, for other signs they appear more flexible. Especially the location of the sign is often loosely specified as *neutral space*, which is very broad. Figure 15 shows an example of this for the sign FEE-A that signifies *fairy*. It represents the casting of a wand, but the precise location or even direction of the virtual wand is not very restrictive. In this case, the query is a correct variation of the sign but the location and the direction of the movement are different and the resulting similarity between query and key is low (0.31).
4. *Label-Noise from Invalid User Input* represents a final challenge for SignBuddy's performance. A subset of recordings deviates entirely from the prompted sign, comprising two distinct types of noise. The first type is non-sign gestures. Users (signers and non-signers) occasionally perform arbitrary movements unrelated to any valid sign (e.g., erratic arm motions), generating inputs with no semantic correspondence to the VGT corpus. In the second type, the users deliberately execute signs different from the prompted target to evaluate system robustness. While these recordings may exhibit high recording quality (e.g., clear keypoints), their intentional label mismatch introduces noise for evaluation. Both types of label-noise create a misalignment between input data and ground-truth labels, artificially inflating error rates. In an open data-collection, which is performed largely without supervision, this is unavoidable: participants may sometimes be tempted to challenge the system.

5.3.2 Insights

Since keypoint quality has an impact on our model accuracy, it is crucial that keypoint estimators are improved in two aspects: depth estimation should become more robust, and keypoint estimators should incorporate some form of temporal tracking. Although such errors could be mitigated by a more robust keypoint estimator, no lightweight, state-of-the-art solution is currently as widely supported as MediaPipe. Nevertheless, simply adjusting the recording phase in SignBuddy (see Sect. 6) could yield additional improvements. Additionally, since certain signs have some leeway in their execution (e.g., the orientation of the aforementioned FEE/fairy), recording multiple exemplary sign variations and performing retrieval using *k*-Nearest-Neighbor classification could make retrieval of such signs more robust. Having recordings of several variations of signs could also benefit linguistics research. Finally, these findings underscore the need for robust input validation mechanisms to filter invalid samples before using them for training or testing. In a future deployment scenario, filtering out invalid inputs can also enhance the system's credibility.

[13] Although MediaPipe's output is three-dimensional, the depth dimension is not accurate for robust sign language classification.

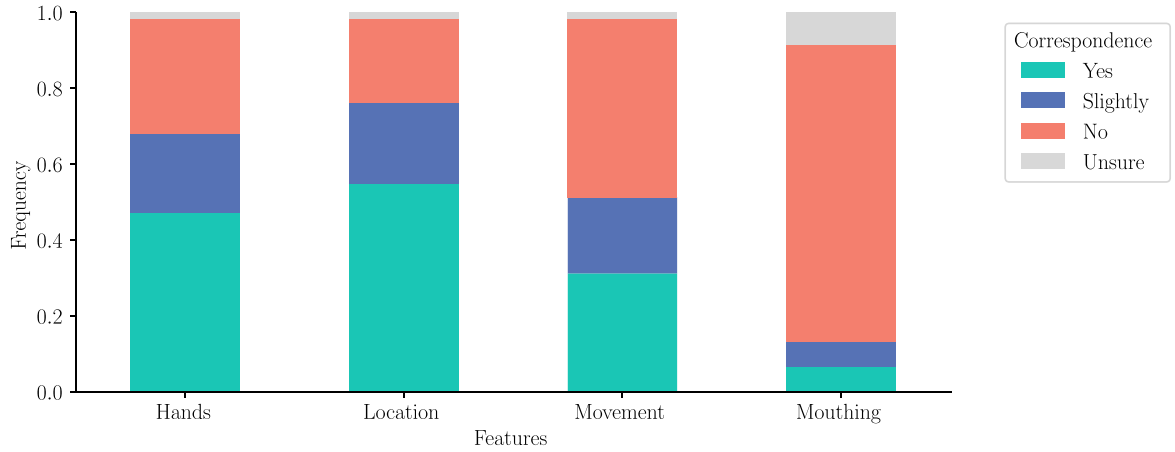


Fig. 9 Categorical user feedback of the signers comparing features of predicted and performed signs

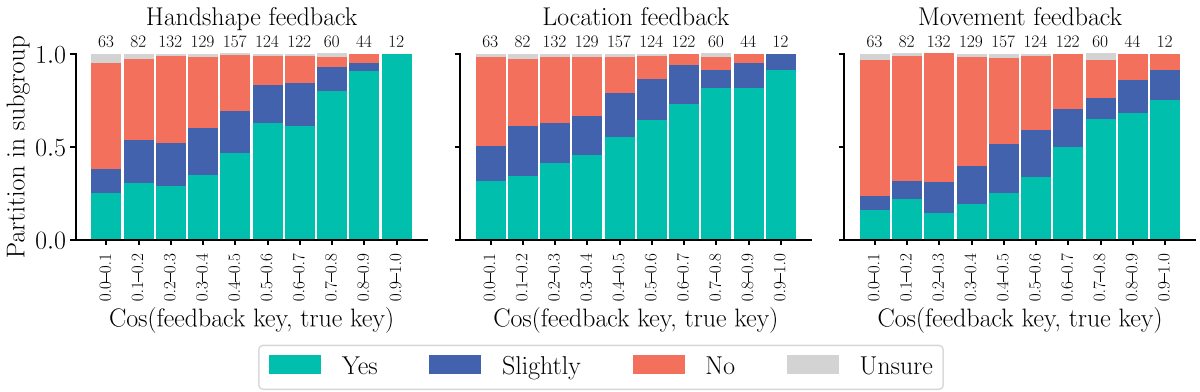


Fig. 10 Categorical user feedback of the signers comparing features of the feedback and key signs. These signs are binned according to the cosine similarity of the two sign embeddings. We visualise three sign parameters, (Left) handshape feedback, (Middle) location and (Right) movement

5.4 Collected feedback

5.4.1 Analysis

As mentioned in Sect. 3.2, users are presented with an optional short questionnaire for the highest-ranked erroneous prediction. We ask them to compare key parameters of their performed sign—hand shape, location, movement, and mouth pattern—to those of the predicted sign that is selected for feedback, i.e., the sign that is most similar to the query when this differs from the key, and the second most similar sign otherwise. In most cases, this feedback sign has a high measured similarity with the query sign according to our model. For samples without strong artefacts like the ones discussed in the previous section, the provided feedback allows us to align measured similarity with percieved similarity and gives insight into the relative importance of the different sign parameters in the model's embedded representations. If, for example, two signs with very different hand shapes have a high measured similarity, these handshapes were not important in the embedding.

Figure 9 visualises these results. The questionnaire is optional: out of the 1868 collected samples, 1101 have associated feedback. Feedback was mostly given by signers (954 out of the 1101 collected feedback samples). The purpose of this feedback is to analyse whether the quantitative (dis)similarities captured in the embeddings align with perceived dissimilarities along different sign parameters. Insights obtained from this may again help to identify routes for future improvement. Because the non-signer subset is relatively small (147 responses, 13.4% of feedback) and may be less reliable for technical phonological judgements, Figs. 9 and 10 only visualise signer responses.

Remark that the results in Figs. 9 and 10 are derived from user assessments and do not directly quantify model performance. Instead, they provide insight into how a set of users perceive model confusions and evaluate model errors. In general, the parameters that are most often perceived as similar between the two compared signs are the hand shapes and location of the sign execution. Movement features were rated as *Similar* in 31.2% of the cases. In contrast, the mouth is indicated as *Not similar* 78.1% of the time. This was

expected because detailed mouth keypoints are currently not included in the model. Past experiments with the inclusion of detailed facial keypoints have not been successful. One possible explanation for this is that the facial keypoint extraction in Mediapipe is not sufficiently accurate to capture any of the finer details of mouth movements beyond the mere opening and closing of the mouth. Also, many mouth movements reflect those of spoken words in the regional language, so the discriminating mouth features may be very language specific (explaining why adding mouth patterns to a model trained on ASL does not aid in VGT sign recognition). Another possible cause is that not all signers use mouth movements. This may also have been the case in the training data.

Figure 10 considers the user feedback for hand shape, location, and movement for different intervals of cosine similarity between the feedback key and the true key. First, we see that similarity scores of our model in the highest ranges almost always correspond to perceived similarity across all three sign parameters. Second, according to user feedback, the hand shape often remains similar for many sign pairs with lower measured similarity for the hands and the location, both dropping rapidly for similarity scores below 0.7. For movement, the reported similarities drop off much more quickly. This means that identical movement patterns are less important to achieve very large similarities according to the model.

Based on this initial sample, we can conclude that the similarities reported by our model are relatively well aligned with perceived similarities for hand shapes and hand locations, whereas for movement there are at times clear differences. This suggests that our model may not be sensitive to differences in movement, and focuses primarily on hand shape and location to classify signs. Even in cases where the similarity between the feedback key and true key is low, i.e., the signs are very different according to our model, the model still tends to favour signs which have a similar hand shape and location.

5.4.2 Insights

Based on user feedback, performance could be improved by incorporating accurate mouth pattern recognition to distinguish between minimal pairs.[14] Possibly, the mouth pattern recognition model should be language-specific, because mouthings used in a sign language are often linked to the regional spoken language (e.g., ASL-English, VGT-Dutch). Hand shape and location are often perceived to be correct, even when the feedback sign is dissimilar from the key sign (according to our model), and movement may not be captured to its full extent by the model.

6 From SignBuddy to the first fully scalable sign-to-text dictionary

One year after the release of SignBuddy, we added sign-to-text search to the VGT dictionary. Although the original recall@6 criterion of 0.66 was not met on the full collection of proficient signer samples, Fig. 8 tells a different story: samples with more recorded frames correlate with improved retrieval results, often exceeding the established threshold. This demonstrates that small alterations to the recording pipeline can effectively increase the success rate of searches.

The first—and most important, yet uncomplicated—improvement is the modification of the sign recording process. Currently, the sign video is buffered locally and processed by MediaPipe post-recording. This processing is performed entirely on the user's device to ensure no personal data is transmitted over the internet. This adjustment shifts the bottleneck from MediaPipe processing to the user's webcam, allowing more frames to be captured per recording.

The second enhancement is an alignment check: before pressing the record button, users must ensure their head and elbows are visible in the frame. MediaPipe processes a few frames in real time before recording to enforce this alignment check. Together, these two alterations largely resolve the hurdles described in cases 1 and 2 of section Sect. 5.3.1, poor keypoint quality and temporal instability. Remaining issues, such as missing hands, could be addressed with a more reliable keypoint estimator. However, MediaPipe remains the most widely accessible option, has tolerable processing times and is likely to enjoy the longest support. For this long-term perspective, we chose not to replace the keypoint estimator itself.

A third and final modification involves shifting the visualisation from Top-6 to Top-9 results. Hassan et al. [52] showed that the placement of the desired result on the first results screen strongly influences users' perceptions of dictionary search systems. By modestly increasing the number of signs displayed, at the cost of slightly smaller video thumbnails, we aim to enhance user satisfaction and foster a more positive attitude towards SLP technology. On our collected data, this final adjustment has a small but notable effect on retrieval performance, as illustrated in Fig. 8. However, as the SignBuddy dataset is as yet too small to make highly reliable performance claims, it builds in some additional margin.

As a result of these three improvements, the first publicly available sign-to-text dictionary search system can now

[14] Two signs form a minimal pair if they differ in only one articulatory parameter, but all other parameters are identical.

be tested at woordenboek.vlaamsegebarentaal.be/direct-record, marking a significant step towards accessible, user-centered sign language technology. Obviously, there is still room for improvement to the model, so the collaboration between VGTC and UGent continues.

7 Future work

This section summarises the insights gained from our data analysis and provides three main avenues for improving SignBuddy's performance. It can serve as a guideline for future research efforts that would want to set up similar applications for different sign languages.

Our first finding—while highly anticipated—is quantitatively validated within this work: in-domain training significantly impacts application performance. As shown in Sect. 5.1, the model achieves reliable classification for signs embedded in dense clusters within ASL-Citizen but underperforms on linguistically distinct signs. Notably, we caution against conflating same-language pretraining with in-domain training. For SignBuddy—a tool for VGT—pretraining on the largest available VGT dataset [37] yields poor performance due to domain mismatch (e.g., sections from continuous signing or signers positioned at an angle) and Zipfian class distribution (i.e., very few samples for most of the signs). Instead, we advocate for in-domain training, where datasets align with the evaluation task in both represented language and sourcing validity (e.g., webcam-captured signs with similar variation to the real-world dictionary of evaluation). The latter is exemplified by ASL-Citizen's web-sourced data, which enhances accuracy by mirroring real-world deployment conditions.

From the qualitative analysis in Sect. 5.2, we can draw the following conclusions. The language mismatch from ASL training to VGT inference is a non-issue: even though signs in the dictionary are mapped to a sparse region of the ASL embedding space, the performance does not suffer. Therefore, language specific fine-tuning should not be the main priority for model improvement. What is more critical is the quality of query and key signs: as we saw in Sect. 5.1, the performance is higher for signers than for non-signers. Moreover, many of the failure cases of our model are due to keypoint mistakes, as illustrated in Sect. 5.2 and further categorised in Sect. 5.3.

The quality limitations identified in Sect. 5.3 largely stem from MediaPipe's inconsistent keypoint detection, particularly under real-world conditions (e.g., occlusion, rapid motion). However, these artefacts do not negate the rationale for selecting keypoint-based input—namely, effective anonymization and computational efficiency through dimensionality and noise reduction. To address detection errors, we propose augmenting the prediction pipeline with an out-of-distribution (OOD) check. As evidenced in Sect. 5.1, recordings with low accuracy scores map to sparser regions of the pretrained embedding space, likely reflecting anomalous inputs (e.g., incomplete keypoints, erratic trajectories). By flagging such OOD samples during inference, the system can prompt users to re-record signs more mindfully, thereby mitigating: poor keypoint quality *(Case 1)*; temporal instability *(Case 2)*; label-noise *(Case 4)*. While more robust keypoint extractors exist [55, 56], these are generally more compute-heavy than MediaPipe and therefore not suited for edge deployment on many of our users' devices.

Another important factor related to MediaPipe's keypoint prediction accuracy is linked to SignBuddy's one-shot classification approach. If the keypoint predictions for the *key* sign are particularly bad, all *queries* that do not exhibit the same artefacts will by default obtain bad similarity scores. An example of such a sign is RIJ-A, depicted in Fig. 11. In such cases, which can be automatically detected and flagged, it is sensible to (manually) correct the keypoints of the key sign recording.

These analyses lead us to selecting these two main avenues for improving SignBuddy's performance: (1) correcting erroneous keypoint predictions in *key signs*; (2) improving the training set variety to populate sparse regions of the model's embedding space. Moreover, we will thoroughly analyse and clean the collected data to reduce label noise to obtain a more accurate estimate of the true performance of the search functionality.

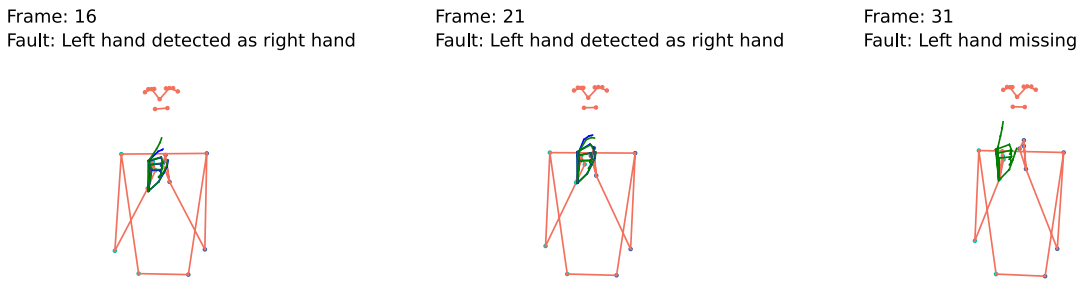


Fig. 11 The key for RIJ-A contains several artefacts due to MediaPipe's mistakes

8 Reflections on the co-creation process

Our collaboration confirms that co-creation is not merely the presence of a shared goal or a formal agreement; it is shaped by the practical conditions under which partners work together. In reviewing the process through the lens of the five lessons proposed by De Meulder et al. [2], we observed that some principles were foundational, while others emerged as ongoing challenges rather than steps to be "checked off."

The most decisive factor was *Lesson 5: redistributing power through DHH leadership.* Beginning with DHH leadership—rather than gradually incorporating it—created the conditions for genuine co-creation. Decisions taken early on by VGTC shaped the project's goals, tone, community approach, and accessibility measures. This confirmed that redistributing power is not the final step of co-creation but the structural starting point.

Lessons 2 and 3—managing expectations and dismantling structural ableism—proved to be less about one-off project planning and more about continuous negotiation. Transparent communication, repeated clarifications about what AI can and cannot do, and the need to align working capacities across partners required persistent attention. These lessons were not abstract principles: they directly prevented misunderstandings, unrealistic expectations, and friction in workload distribution.

Diversifying user participation (Lesson 4) emerged as a pragmatic necessity. Recruitment through public-facing channels rather than the personal networks of DHH researchers reduced community fatigue and prevented the over-reliance on specific people or groups that often accompanies small linguistic communities. This diversification was not an added benefit but a corrective to common patterns of extraction described by De Meulder et al. [2].

Finally, our collaboration highlighted that *Lesson 1—recognising and resourcing invisible labour*—is the area where co-creation most easily slips into imbalance. Community mediation, trust-building, and accessibility decisions were consistently carried by DHH partners. This remains the lesson that demands the most vigilance moving forward.

9 Conclusion

This paper describes the conceptualisation, realisation, and evaluation of SignBuddy, emphasising challenges inherent to the real-world deployment of sign language technologies. We detail a co-creation methodology grounded in a close collaboration between VGTC and Ghent University, ensuring the tool's alignment with user needs and linguistic authenticity. We demonstrate how the co-creation principles—starting from DHH leadership, and persistent attention to managing expectations, dismantling structural ableism, diversifying user participation and recognising invisible labour—can lead to the realisation of sign language processing (SLP) technologies that genuinely serve the DHH community. Guided by seven core principles—(1) accessibility, (2) inclusivity, (3) usability and clarity, (4) privacy, (5) user engagement, and (6) data collection variety and (7) collected data pertinence—we prioritised both ethical and pragmatic design (principles 1–5) and empirical rigour (principles 6–7). These principles ensured that user contributions directly informed technical insights for the applications to come.

The resulting application, SignBuddy, is a data collection tool. Its developement enabled the release of the first ISLR model that can search through over ten thousand signs in the VGT-Dutch online dictionary. The underlying model achieves state of the art performance on ASL-Citizen, and can be applied in a one-shot setting to data from a different language without language-specific fine-tuning.

Furthermore, we provide insights derived from a real-world sign language application. We discover several potential research paths by conducting quantitative and qualitative inspections of the gathered data. By bridging co-design with computational innovation, this work advances the broader objective of sign language technology research: to develop equitable tools that empower DHH communities through reliable, accessible applications.

The tangible outcome of this collaboration is the first fully scalable and publicly available AI-based sign-to-text dictionary search system, transforming an academic prototype into an accessible public resource. More broadly, this work illustrates that technological innovation in SLP does not solely depend on algorithmic advances but on equitable partnerships that bridge linguistic, cultural, and technical expertise. By grounding research in co-creation, we ensure that future systems evolve in dialogue with the communities they aim to support—laying the foundation for a new generation of inclusive, transparent, and socially sustainable sign language technologies.

Appendix A: PoseFormer pretraining

Underlying SignBuddy's search system is the PoseFormer, an ISLR model that combines convolutional neural networks with self-attention [18]. This model was chosen for three reasons:

- the inputs to this model are keypoint sequences obtained with MediaPipe [26]: this aligns with SignBuddy's privacy goal,

- the model has an architecture similar to the top solutions of a recent Kaggle competition for MediaPipe-based ISLR [28]: this illustrates that it is a powerful architecture, and
- pre-trained versions of this model on various sign languages are available on the HuggingFace hub.[15] We trained this model in a supervised manner on the ASL-Citizen dataset [54]. We optimise categorical cross-entropy during training, with a batch size of 64 and learning rate of 0.0003. We reduce the learning rate when the validation set accuracy plateaus, and employ early stopping. There are 4 8-head self-attention layers, with a dropout probability of 0.2. Every frame in the keypoint sequence is embedded onto a 160-dimensional embedding space. These values were obtained after extensive hyperparameter tuning.

Using these settings, we achieve competitive results on ASL-Citizen, outperforming the previous state of the art by 12% Recall@1, 0.1 MRR and 0.079 nDCG (these metrics are explained in Appendix B). This is illustrated in Table 2.

Appendix B: Evaluation metrics

Multiple evaluation metrics are employed to ensure robustness across diverse experimental contexts. Among these, we emphasise mean Recall@K for $K \in [1, 2, 4, 5, 6, 10]$. Recall@$K$ quantifies the frequency with which the ground-truth sign class appears in the model's top K predictions. The selected K values serve distinct purposes: $K = 1 - 5 - 10$ aligns with standard benchmarks for state-of-the-art comparisons, while less frequently used K values (e.g. K = 6) reflect the SignBuddy-user experience. However, the interpretation of the metric remains the same. The recall at any given K reflects how many times the correct prediction was presented within a given set of K predictions.

Furthermore, two ranking metrics are employed: mean reciprocal rank (MRR) and normalised discounted cumulative gain (nDCG). MRR measures the inverse of the first correct prediction's rank. Let r_i be the rank for the i^{th} test example, that is, the one-based index of the ground truth label in the ordered list of model predictions. Then, for a set of N test examples,

Table 2 The PoseFormer outperforms the I3D baseline (ASL Citizen) on the pretraining task for all considered metrics

Model	↑ MRR	↑ nDCG	↑ Rec@1	↑ Rec@5	↑ Rec@10
Poseformer	0.833	0.870	0.751	0.932	0.955
I3D [54]	0.733	0.791	0.631	0.861	0.909

[15] huggingface.co/signon-project.

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i}. \tag{A1}$$

nDCG is similar to MRR, but it considers the ranks of all correct predictions, favouring relevant results that occur earlier in the ordered list of model predictions. Since in our case search results are either correct or incorrect, we define the relevance of a search result as a binary value: it is 1 when the prediction $\hat{y}$ is equal to the ground truth label y, and 0 otherwise. The DCG can be computed for a single test example by considering the ordered list of M predictions. For an expected ground truth label y, the DCG is equal to

$$\text{DCG} = \sum_{j=1}^{M}\frac{\mathbf{1}_{\hat{y}_j=y}}{\log_2(j+1)}. \tag{A2}$$

nDCG is the normalised version of DCG. This normalisation is done by dividing the DCG by the ideal DCG (IDCG). IDCG is the DCG of the optimal ranking. In our specific case, with only one relevant item, the IDCG is always equal to one. Therefore the nDCG simplifies directly to DCG:

$$\text{nDCG} = \frac{\text{DCG}}{\text{IDCG}} = \text{DCG}. \tag{A3}$$

We compute the DCG for all N test examples and report the mean.

Given that there is only one relevant item per prediction (its label), the interpretations of MRR and DCG are similar in this scenario. The key difference is that the inverse of the MRR is the harmonic mean of the ranks of all predictions, which provides an alternative view of the results. For both the MRR and the DCG, higher is better and a value of one indicates optimal performance. All three metrics—Recall@K, MRR, and DCG—are used to evaluate both pretraining and dictionary lookup.

Finally, we employ cosine similarity to quantify the angular distance between embedding vectors in the model's latent space. Mathematically, cosine similarity is defined as the dot product of two vectors divided by the product of their magnitude (L2 norms), yielding a normalised measure within the range $[-1, 1]$. Values closer to 1 indicate high similarity, while values approaching -1 signify dissimilarity. In the context of isolated sign recognition, this metric reveals how the model clusters related signs. Formally, the cosine similarity between two vectors x and y is defined as

$$\cos(\mathrm{x}, \mathrm{y}) = \frac{\mathrm{x}\cdot\mathrm{y}}{\|\mathrm{x}\|\|\mathrm{y}\|}. \tag{A4}$$

Appendix C: Examples of failure cases

To understand why the cases below cause MediaPipe and/or our model to fail, it is essential to understand how MediaPipe and SignBuddy work and interact. MediaPipe's pose estimation is a pipeline consisting of several parts. First, given an image, the tool performs object detection to obtain 2D bounding boxes of detected humans. Then, it estimates the body pose, predicting keypoints for, among others, the shoulders, elbows, and wrists. Next, a hand detection model is used to predict bounding boxes near the wrist keypoints, producing up to two hand crops. A second keypoint estimation model, specifically for hand keypoints, is then run on these hand crops. The resulting hand keypoints are combined with the body keypoints to create the complete pose. MediaPipe predicts three coordinates (x, y, and z) for all keypoints. However, the body and hand keypoint models are not trained to accurately predict depth (z) coordinates, and including these in an SLR model typically has a negative impact on performance. Therefore, SignBuddy removes the z coordinate, retaining only the x and y dimensions. Due to the heterogeneity of users' webcams and the lack of knowledge about their intrinsic parameters, this orthogonal projection is the best approximation we can achieve. In reality, webcams have varying resolutions and fields of view, and users can be positioned at a multitude of angles. This introduces noise into the 2D projection transformation, which contributes to many of the failure cases mentioned in this appendix.

When signs are produced slightly higher or lower than the exemplary position, the one-shot classification system fails to assign the correct label to the query. This can be seen in Figs. 12 and 13. A native signer may be able to correctly classify the query, but our machine learning model fails to do so. Similar failures can occur when hands are slightly rotated.

Our system relies on 2D keypoint estimation, and small differences in 3D rotation can lead to large differences in the 2D projection and the resulting 2D keypoint trajectories. Figure 12 illustrates this with an example for the sign KELDER-A, Fig. 14 for the sign BENZINE-E, and Fig. 15 for the sign FEE-A. When the user is not facing the camera, but sitting at a slight angle, the location of the sign may appear wrong due to the naive 2D projection taking place. This can also cause misclassifications, such as the one shown in Fig. 12 (Fig. 13).

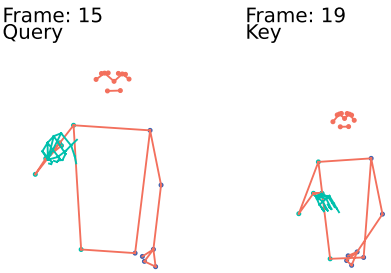


Fig. 12 The right hand in this query for KELDER-A is rotated differently than in the query. This may cause the sign to have a different meaning according to PF-ASL, and therefore lead to misclassification. Since the sign is produced in neutral space, the hand location in the query is correct; however, PF-ASL has only seen one example (the key) with a different location, also confusing the model

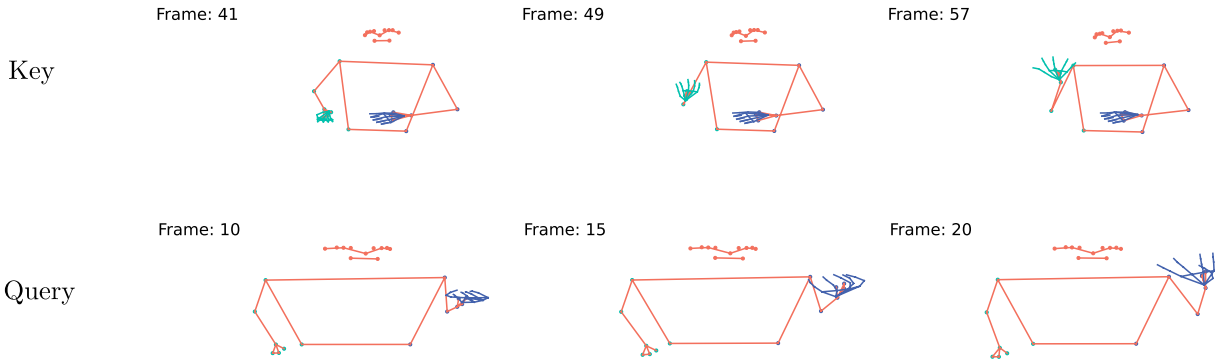


Fig. 13 This query for BOWLING-A is not executed in the same location as the key: the signing location is higher compared to in the key. Note that the pose in the query sign appears stretched: this is likely due to a smaller field of view of the user's webcam. This discrepancy also causes issues with the 2D projection of the keypoints

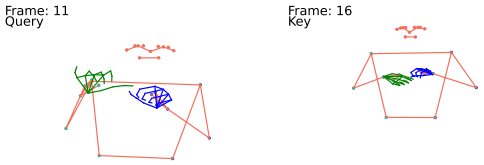


Fig. 14 The hands in this query for BENZINE-E are rotated differently, similar to KELDER-A in Fig. 12

When the left and right hands interact, MediaPipe may fail to correctly predict keypoints for the occluded hand: it may either not detect the hand and not predict any keypoints, or it may detect the left hand as the right hand or vice versa. This can be seen in Fig. 16, which shows the hands for a selection of frames for the sign RIJ-A (see also: Fig. 11). In the first and last frame, the left and right hands are properly separated. In the middle frames, the hand interaction causes MediaPipe to predict erroneous keypoints or no keypoints at all.

Fig. 15 (Left) This query for FEE-A displays the correct hand shapes and hand movement. The location and the direction of the movement are also within allowed ranges but the resulting keypoint representations are considerably different. (Right) Phonological annotation of FEE-A in the VGT dictionary. The right section of the image illustrates the required hand shapes and the very relaxed location requirements (neutral zone), as indicated in the dictionary

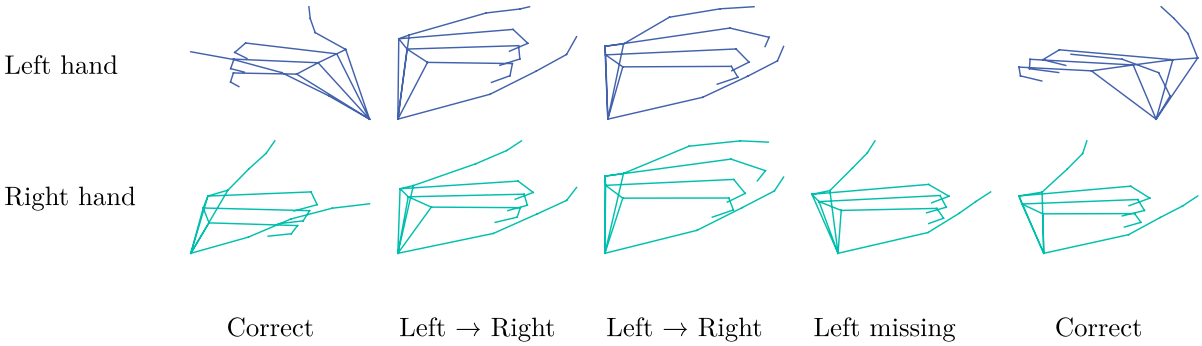


Fig. 16 Due to the interacting hands in RIJ-A, MediaPipe fails to correctly predict keypoints for the left hand. We discuss the frames in chronological order from left to right. In the first frame, MediaPipe is able to separate the hands, and correctly predicts separate keypoints for the left and right hand. In the second and third frames, both hands are present in the detection made by MediaPipe; it confuses the left and right hand, and predicts the right hand keypoints for the left hand. In the fourth frame, MediaPipe does not detect the left hand and predicts no keypoints. In the last frame, MediaPipe again correctly separates the hands and predicts separate keypoints for the left and right hand

Acknowledgements For both partners, a large part of the work in this paper was funded by the "Wat Gebaar Jij?!" project. This project funding was assigned by the Flemish government in the context of the program for citizen science for AI—amai! (amai.vlaanderen). The authors thank all those who recorded and uploaded signs to SignBuddy for their contribution to this project. They also thank the IDLab-AIRO members who contributed to the original validation set.

Author Contributions Conceptualization: all authors; Methodology: all authors; Formal analysis and investigation: T.V., C.B., M.D. and J.D. ; Writing - original draft preparation: T.V. ; Writing - review and editing: T.V., C.B., M.D. and J.D.; Funding acquisition: C.B., H.D., M.D. and J.D.; Resources: all authors; Supervision: H.D., J.D.

Data Availability The data collected for this study is available upon request. Requests should be directed to Joni Dambre (joni.dambre@ugent.be).

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Lepp, L., Shterionov, D., Sisto, M.D., Chrupała, G.: Co-creation for sign language processing and machine translation (2025). https://arxiv.org/abs/2503.01553
2. De Meulder, M., Van Landuyt, D., Omardeen, R.: Lessons in co-creation: the inconvenient truths of inclusive sign language technology development. arXiv preprint arXiv:2408.13171 (2024)
3. Yin, K., Atwell, K., Hochgesang, J.A., Alikhani, M.: The importance of including signed languages in natural language processing. In: Way, A., Leeson, L., Shterionov, D. (eds.) Sign Language Machine Translation 2024 Jul 26, pp. 73–87. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-47362-3_3
4. De Coster, M., Shterionov, D., Van Herreweghe, M., Dambre, J.: Machine translation from signed to spoken languages: State of the art and challenges. Univer. Access Inf. Soc. **23**(3), 1305–1331 (2024)
5. Fox, N., Woll, B., Cormier, K.: Best practices for sign language technology research. Univer. Access Inf. Soc
6. De Meulder, M.: Is "good enough" good enough? Ethical and responsible development of sign language technologies. In: Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL), pp. 12–22 (2021)
7. Wolfe, R., Mcdonald, J.C., Efthimiou, E., Fotinea, E., Picron, F., Van Landuyt, D., Sioen, T., Braffort, A., Filhol, M., Ebling, S.: The myth of signing avatars. In: 1st International Workshop on Automatic Translation for Signed and Spoken Languages (2021)
8. Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T.: Sign language recognition, generation, and translation: an interdisciplinary perspective. In: Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility (2019)
9. Shterionov, D., Vandeghinste, V., Saggion, H., Blat, J., De Coster, M., Dambre, J., Heuvel, H., Murtagh, I., Leeson, L., Schuurman, I.: The signon project: a sign language translation framework. In: 31st Meeting of Computational Linguistics in The Netherlands (CLIN 31) (2021)
10. Picron, F., Van Landuyt, D., Omardeen, R., Efthimiou, E., Wolfe, R., Fotinea, S.-E., Goulas, T., Tismer, C., Kopf, M., Hanke, T.: The easier mobile application and avatar end-user evaluation methodology. In: Proceedings of the LREC-COLING 2024 11th Workshop on the Representation and Processing of Sign Languages: Evaluation of Sign Language Resources, pp. 276–281 (2024)
11. Tilburg University: CoCoS: From Collaboration to Co-creation for Sign Language Machine Translation Research (2025). https://www.tilburguniversity.edu/about/digital-sciences-society/projects/cocos. Accessed 3 Feb 2025
12. University of Surrey: SignGPT (2025). https://www.surrey.ac.uk/news/signgpt-project-awarded-ps845m-build-sign-language-ai-model-deaf-community. Accessed 3 Feb 2025
13. Hill, J.: Do deaf communities actually want sign language gloves? Nat. Electron. **3**(9), 512–513 (2020)
14. Van Herreweghe, M., Vermeerbergen, M., De Weerdt, K., Van Mulders, K.: Woordenboek Nederlands–Vlaamse Gebarentaal/Vlaamse Gebarentaal–Nederlands (online) (2004) https://woordenboek.vlaamsegebarentaal.be/
15. Hu, H., Zhao, W., Zhou, W., Li, H.: Signbert+: hand-model-aware self-supervised pre-training for sign language understanding. IEEE Trans. Pattern Anal. Mach. Intell. **45**(9), 11221–11239 (2023)
16. Wong, R., Camgoz, N.C., Bowden, R.: Sign2gpt: leveraging large language models for gloss-free sign language translation. In: The 12th International Conference on Learning Representations
17. Chen, Y., Wei, F., Sun, X., Wu, Z., Lin, S.: A simple multi-modality transfer learning baseline for sign language translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5120–5130 (2022)
18. Holmes, R., Rushe, E., De Coster, M., Bonnaerens, M., Satoh, S., Sugimoto, A., Ventresque, A.: From Scarcity to Understanding: Transfer Learning for the Extremely Low Resource Irish Sign Language. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2008–2017 (2023)
19. Moryossef, A., Tsochantaridis, I., Dinn, J., Camgoz, N.C., Bowden, R., Jiang, T., Rios, A., Muller, M., Ebling, S.: Evaluating the immediate applicability of pose estimation for sign language recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3434–3440 (2021)
20. De Coster, M., Rushe, E., Holmes, R., Ventresque, A., Dambre, J.: Towards the extraction of robust sign embeddings for low resource sign language recognition. arXiv preprint arXiv:2306.17558 (2023)
21. Sincan, O.M., Camgoz, N.C., Bowden, R.: Using an llm to turn sign spottings into spoken language sentences. arXiv preprint arXiv:2403.10434 (2024)
22. Desai, A., De Meulder, M., Hochgesang, J.A., Kocab, A., Lu, A.X.: Systemic biases in sign language ai research: A deaf-led call to reevaluate research agendas. arXiv preprint arXiv:2403.02563 (2024)
23. Camgoz, N.C., Hadfield, S., Koller, O., Ney, H., Bowden, R.: Neural sign language translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7784–7793 (2018)

24. Papadimitriou, K., Potamianos, G.: Sign language recognition via deformable 3d convolutions and modulated graph convolutional networks. In: ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023).https://doi.org/10.1109/ICASSP49357.2023.10096714
25. Chen, Y., Zuo, R., Wei, F., Wu, Y., Liu, S., Mak, B.: Two-stream network for sign language recognition and translation. Adv. Neural. Inf. Process. Syst. **35**, 17043–17056 (2022)
26. Grishchenko, I., Bazarevsky, V.: MediaPipe holistic—simultaneous face, hand and pose prediction, on device. Posted by Research Engineers, Google Research (2020)
27. Cao, Z., Simon, T., Wei, S.-E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299 (2017)
28. Chow, A., Cameron, G., Sherwood, M., Culliton, P., Sepah, S., Dane, S., Starner, T.: Google—Isolated Sign Language Recognition. Kaggle (2023). https://kaggle.com/competitions/asl-signs
29. Chow, A., Cameron, G., Georg, M., Sherwood, M., Culliton, P., Sepah, S., Dane, S., Starner, T.: Google-American sign language fingerspelling recognition (2023) https://www.kaggle.com/competitions/asl-fingerspelling
30. Koller, O., Zargaran, O., Ney, H., Bowden, R.: Deep sign: hybrid cnn-hmm for continuous sign language recognition. In: Proceedings of the British Machine Vision Conference 2016 (2016)
31. Koller, O., Zargaran, S., Ney, H.: Re-sign: re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
32. Ye, Y., Tian, Y., Huenerfauth, M., Liu, J.: Recognizing American sign language gestures from within continuous videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 2064–2073 (2018)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv, Neural Inf. Process. Syst. **30**, 6000-6010 (2017)
34. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. **28**(4), 594–611 (2006)
35. Wang, F., Li, C., Zeng, Z., Xu, K., Cheng, S., Liu, Y., Sun, S.: Cornerstone network with feature extractor: a metric-based few-shot model for Chinese natural sign language. Appl. Intell. **51**, 7139–7150 (2021)
36. De Coster, M., Dambre, J.: Querying a sign language dictionary with videos using dense vector search. In: 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), pp. 1–5. IEEE (2023)
37. Van Herreweghe, M., Vermeerbergen, M., Demey, E., De Durpel, H., Nyffels, H., Verstraete, S.: Het Corpus VGT. Een digitaal open access corpus van videos and annotaties van Vlaamse Gebarentaal, ontwikkeld aan de Universiteit Gent i.s.m. KU Leuven (2015). https://www.corpusvgt.be
38. Starner, T., Forbes, S., So, M., Martin, D., Sridhar, R., Deshpande, G., Sepah, S., Shahryar, S., Bhardwaj, K., Kwok, T., et al.: Popsign asl v1. 0: an isolated American sign language dataset collected via smartphones. Adv. Neural Inf. Process. Syst. **36**, 184–196 (2024)
39. Zhang, Y., Min, Y., Chen, X.: Teaching chinese sign language with a smartphone. Virtual Real. Intell. Hardware **3**(3), 248–260 (2021)
40. Adamo-Villani, N., Wright, K.: Smile: an immersive learning game for deaf and hearing children. In: ACM SIGGRAPH 2007 Educators Program, p. 17 (2007)
41. Zafrulla, Z., Brashear, H., Presti, P., Hamilton, H., Starner, T.: Copycat: an american sign language game for deaf children. In: Face and Gesture 2011, pp. 647. IEEE Computer Society (2011)
42. Bansal, D., Ravi, P., So, M., Agrawal, P., Chadha, I., Murugappan, G., Duke, C.: Copycat: using sign language recognition to help deaf children acquire language skills. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–10 (2021)
43. Brosens, C., Janssens, M., Verstraete, S., Vandamme, T., De Durpel, H.: Moving towards a functional approach in the flemish sign language dictionary making process. In: Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources, pp. 24–28 (2022)
44. Fink, J., Poitier, P., André, M., Meurice, L., Frénay, B., Cleve, A., Dumas, B., Meurant, L.: Sign language-to-text dictionary with lightweight transformer models. In: 32nd International Joint Conference on Artificial Intelligence, IJCAI 2023, pp. 5968–5976. International Joint Conferences on Artificial Intelligence (2023)
45. Cassidy, S., Crasborn, O., Nieminen, H., Stoop, W., Hulsbosch, M., Even, S., Komen, E., Johnston, T.: Signbank: Software to support web based dictionaries of sign language. In: Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018) (2018)
46. Vermeerbergen, M., Van Herreweghe, M.: Looking back while moving forward: the impact of societal and technological developments on Flemish sign language lexicographic practices. Int. J. Lexicogr. **31**(2), 167–195 (2018)
47. Sutton, V.: Lessons in SignWriting. SignWriting Press (2022)
48. Stokoe Jr, W. C.: Sign language structure: an outline of the visual communication systems of the American deaf studies and deaf education, **10**(1), 3-37 (2005)
49. Battison, R.: Lexical Borrowing in American Sign Language. Linstok Press, Silver Spring (1978)
50. AMAI!Vlaanderen: Wat gebaar jij?! (2025). https://amai.vlaanderen/projecten/project3-gebaar. Accessed 15 Feb 2025
51. Goldin-Meadow, S., Mayberry, R.I.: How do profoundly deaf children learn to read? Learn. Disab. Res. Pract. **16**(4), 222–229 (2001)
52. Hassan, S., Alonzo, O., Glasser, A., Huenerfauth, M.: Effect of sign-recognition performance on the usability of sign-language dictionary search. ACM Trans. Access. Comput. (TACCESS) **14**(4), 1–33 (2021)
53. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
54. Desai, A., Berger, L., Minakov, F., Milano, N., Singh, C., Pumphrey, K., Ladner, R., Daumé III, H., Lu, A.X., Caselli, N., et al.: Asl citizen: a community-sourced dataset for advancing isolated sign language recognition. Adv. Neural Inf. Process. Syst. **36**, 76893-76907 (2024)
55. Dong, H., Chharia, A., Gou, W., Vicente Carrasco, F., Torre, F.D.: Hamba: single-view 3d hand reconstruction with graph-guided bi-scanning mamba. Adv. Neural. Inf. Process. Syst. **37**, 2127–2160 (2025)
56. Pavlakos, G., Shan, D., Radosavovic, I., Kanazawa, A., Fouhey, D., Malik, J.: Reconstructing hands in 3d with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9826–9836 (2024)