Data Reference and Quality Guidelines

1 Wire Format (UDP JSON v4/v5)

The SDK sends its public stream over UDP as UTF-8 JSON datagrams. Two schemas are currently in use: omniglove.stream.v4 (default, mode 0) and omniglove.stream.v5 (POSE, mode 1). Each datagram carries one complete JSON sample, with a maximum payload of 65,507 bytes. This applies to Linux. For schema field definitions, mode constraints, and publishing policies, see Sections 2.3 and 3.1.3 of the **OmniGlove SDK User Guide**.

1.1 Receive and Validate Data

At a minimum, the receiver should:

1. Decode each datagram as UTF-8, then check the schema (omniglove.stream.v4 or v5) and the stream field.
2. Use components, mapped, flags, and field presence to determine which fields are valid. Sections that are not selected are omitted from JSON.
3. Non-finite floating-point values are encoded as `null`, and so are invalid device indices. `Null` values are not replaced with zero when read. Do not treat `null` or zero as a valid measurement.
4. Use `source_sequence` to identify new measurements and `publish_sequence` to check the publishing sequence. Monitor drops and continuity errors through the SDK status and `gap_count`.

1.2 Transport Limitations

- The SDK currently sends over IPv4 UDP in non-blocking mode.
- UDP provides no acknowledgment, retransmission, authentication, or encryption. If you need reliable delivery or a security boundary, provide it in your deployment network or client application.
- A UDP send failure on one stream affects only that stream's status and statistics. It does not stop other streams or device connections.
- `OnNewSample` attempts to publish every sample that enters the publishing queue. The queue is bounded. When it is full, new samples are dropped, and the drops appear in the sequence numbers, gap count, and statistics.

- `FixedRateLatest` publishes the latest sample at the configured interval, so the same `source_sequence` may be published more than once.

2 Tactile Data

2.1 Tactile Matrix Layout

Tactile data comes from piezoresistive arrays that cover the five fingers and the palm, with 746 active sensing points per hand. The figure below shows where each sensing point is located on the glove.

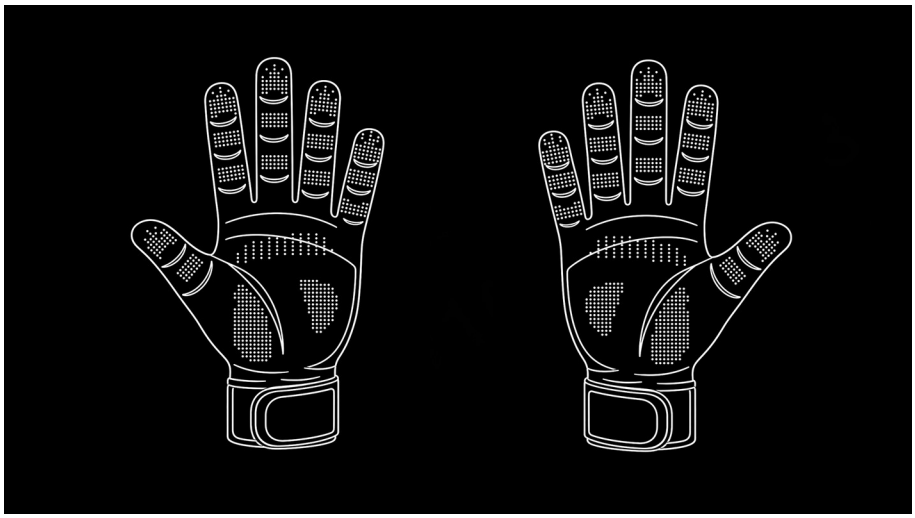


Figure 1

2.2 Data Description

Tactile data is grouped by finger region. Each region is a rectangle in row-major order. Positions in the rectangle that belong to the region carry the force values (in g) of active sensing points. Positions marked 0 are not output. The table and layouts below show each region and the output order of its sensing points.

Table 1

Field	Type	Array Length	Active Points	Region
thumb	List[uint8_t]	84	66	Thumb, 12 × 7
index	List[uint8_t]	112	94	Index finger, 16 × 7

Field	Type	Array Length	Active Points	Region
middle	List[uint8_t]	112	93	Middle finger, 16 × 7
ring	List[uint8_t]	112	93	Ring finger, 16 × 7
pinky	List[uint8_t]	112	92	Little finger, 16 × 7
palm	List[uint8_t]	600	308	palm, 30 x 20

Left thumb

```
{0, 0, 0, 1, 0, 0, 0}, \
{0, 2, 0, 3, 0, 4, 0}, \
{0, 5, 6, 7, 8, 0, 0}, \
{9, 10, 11, 12, 0, 13, 14}, \
{15, 16, 17, 18, 19, 20, 21}, \
{22, 23, 24, 25, 26, 27, 28}, \
{29, 30, 31, 32, 33, 34, 35}, \
{36, 0, 0, 0, 0, 37, 38} \
{39, 40, 41, 42, 43, 44, 45}, \
{46, 47, 48, 49, 50, 51, 52}, \
{53, 54, 55, 56, 57, 58, 59}, \
{60, 61, 62, 63, 64, 65, 66} \
```

Right thumb

```
{0, 0, 0, 1, 0, 0, 0}, \
{0, 2, 0, 3, 0, 4, 0}, \
{0, 0, 5, 6, 7, 8, 0}, \
{9, 10, 0, 11, 12, 13, 14}, \
{15, 16, 17, 18, 19, 20, 21}, \
{22, 23, 24, 25, 26, 27, 28}, \
{29, 30, 31, 32, 33, 34, 35}, \
{36, 37, 0, 0, 0, 0, 38} \
{39, 40, 41, 42, 43, 44, 45}, \
{46, 47, 48, 49, 50, 51, 52}, \
{53, 54, 55, 56, 57, 58, 59}, \
{60, 61, 62, 63, 64, 65, 66} \
```

Left index

```
{0, 0, 0, 1, 0, 0, 0}, \
{0, 2, 0, 3, 0, 4, 0}, \
{0, 0, 5, 6, 7, 0, 0}, \
{8, 9, 10, 11, 12, 13, 14}, \
{15, 16, 17, 18, 19, 20, 21}, \
{22, 23, 24, 25, 26, 27, 28}, \
{29, 30, 31, 32, 33, 34, 35}, \
{36, 37, 0, 0, 0, 0, 38} \
{39, 40, 41, 42, 43, 44, 45}, \
```

```
{46, 47, 48, 49, 50, 51, 52}, \
{53, 54, 55, 56, 57, 58, 59}, \
{60, 61, 62, 63, 64, 65, 66} \
{67, 68, 69, 70, 71, 72, 73}, \
{74, 75, 76, 77, 78, 79, 80}, \
{81, 82, 83, 84, 85, 86, 87}, \
{88, 89, 90, 91, 92, 93, 94} \
```

Right index

```
{0, 0, 0, 1, 0, 0, 0}, \
{0, 2, 0, 3, 0, 4, 0}, \
{0, 0, 5, 6, 7, 0, 0}, \
{8, 9, 10, 11, 12, 13, 14}, \
{15, 16, 17, 18, 19, 20, 21}, \
{22, 23, 24, 25, 26, 27, 28}, \
{29, 30, 31, 32, 33, 34, 35}, \
{36, 0, 0, 0, 0, 37, 38} \
{39, 40, 41, 42, 43, 44, 45}, \
{46, 47, 48, 49, 50, 51, 52}, \
{53, 54, 55, 56, 57, 58, 59}, \
{60, 61, 62, 63, 64, 65, 66} \
{67, 68, 69, 70, 71, 72, 73}, \
{74, 75, 76, 77, 78, 79, 80}, \
{81, 82, 83, 84, 85, 86, 87}, \
{88, 89, 90, 91, 92, 93, 94} \
```

Middle(left/right)

```
{0, 0, 0, 1, 0, 0, 0}, \
{0, 2, 0, 3, 0, 4, 0}, \
{0, 0, 5, 6, 7, 0, 0}, \
{8, 0, 9, 10, 11, 0, 12}, \
{13, 14, 15, 16, 17, 18, 19}, \
{20, 21, 22, 23, 24, 25, 26}, \
{27, 28, 29, 30, 31, 32, 33}, \
{34, 35, 0, 0, 0, 36, 37} \
{38, 39, 40, 41, 42, 43, 44}, \
{45, 46, 47, 48, 49, 50, 51}, \
{52, 53, 54, 55, 56, 57, 58}, \
{59, 60, 61, 62, 63, 64, 65} \
{66, 67, 68, 69, 70, 71, 72}, \
{73, 74, 75, 76, 77, 78, 79}, \
{80, 81, 82, 83, 84, 85, 86}, \
{87, 88, 89, 90, 91, 92, 93} \
```

ring(left,right)

```
{0, 0, 0, 1, 0, 0, 0}, \
{0, 2, 0, 3, 0, 4, 0}, \
{0, 0, 5, 6, 7, 0, 0}, \
{8, 0, 9, 10, 11, 0, 12}, \
{13, 14, 15, 16, 17, 18, 19}, \
{20, 21, 22, 23, 24, 25, 26}, \
```

```
{27, 28, 29, 30, 31, 32, 33}, \
{34, 35, 0, 0, 0, 36, 37} \
{38, 39, 40, 41, 42, 43, 44}, \
{45, 46, 47, 48, 49, 50, 51}, \
{52, 53, 54, 55, 56, 57, 58}, \
{59, 60, 61, 62, 63, 64, 65} \
{66, 67, 68, 69, 70, 71, 72}, \
{73, 74, 75, 76, 77, 78, 79}, \
{80, 81, 82, 83, 84, 85, 86}, \
{87, 88, 89, 90, 91, 92, 93} \
```

left pinky

```
{0, 0, 0, 1, 0, 0, 0}, \
{0, 2, 0, 3, 0, 4, 0}, \
{0, 5, 0, 6, 7, 8, 0}, \
{0, 0, 9, 10, 11, 0, 0}, \
{12, 13, 14, 15, 16, 17, 18}, \
{19, 20, 21, 22, 23, 24, 25}, \
{26, 27, 28, 29, 30, 31, 32}, \
{33, 34, 35, 36, 37, 38, 39} \
{40, 41, 42, 43, 44, 45, 46}, \
{47, 48, 49, 50, 51, 52, 53}, \
{54, 55, 56, 57, 58, 59, 60}, \
{61, 62, 63, 64, 65, 66, 67} \
{68, 69, 0, 0, 0, 70, 71}, \
{72, 73, 74, 75, 76, 77, 78}, \
{79, 80, 81, 82, 83, 84, 85}, \
{86, 87, 88, 89, 90, 91, 92} \
```

right pinky

```
{0, 0, 0, 1, 0, 0, 0}, \
{0, 2, 0, 3, 0, 4, 0}, \
{0, 5, 6, 7, 0, 8, 0}, \
{0, 0, 9, 10, 11, 0, 0}, \
{12, 13, 14, 15, 16, 17, 18}, \
{19, 20, 21, 22, 23, 24, 25}, \
{26, 27, 28, 29, 30, 31, 32}, \
{33, 34, 35, 36, 37, 38, 39} \
{40, 41, 42, 43, 44, 45, 46}, \
{47, 48, 49, 50, 51, 52, 53}, \
{54, 55, 56, 57, 58, 59, 60}, \
{61, 62, 63, 64, 65, 66, 67} \
{68, 69, 0, 0, 0, 70, 71}, \
{72, 73, 74, 75, 76, 77, 78}, \
{79, 80, 81, 82, 83, 84, 85}, \
{86, 87, 88, 89, 90, 91, 92} \
```

left palm

```
{0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 0, 0, 0, 5, 6, 7, 8, 9, 10, 11, 12, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 0, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 0, 0, 0, 0, 0, 0, 0},\
```

```
{46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 0, 0, 0, 0, 0, 0, 0, 0},\
{58, 59, 60, 61, 62, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{63, 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 65, 66, 67, 68, 69, 0, 0, 0, 0, 0},\
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 70, 71, 72, 73, 74, 75, 76, 77, 0, 0},\
{0, 78, 79, 0, 0, 0, 0, 0, 0, 0, 80, 81, 82, 83, 84, 85, 86, 87, 88, 0},\
{0, 89, 90, 91, 0, 0, 0, 0, 0, 0, 0, 92, 93, 94, 95, 96, 97, 98, 99, 0},\
{100, 101, 102, 103, 104, 0, 0, 0, 0, 0, 0, 105, 106, 107, 108, 109, 110, 111, 112, 113},\
{114, 115, 116, 117, 118, 119, 0, 0, 0, 0, 0, 120, 121, 122, 123, 124, 125, 126, 127, 128},\
{0, 129, 130, 131, 132, 133, 134, 0, 0, 0, 0, 0, 135, 136, 137, 138, 139, 140, 141, 142},\
{0, 143, 144, 145, 146, 147, 148, 0, 0, 0, 0, 0, 149, 150, 151, 152, 153, 154, 155, 156},\
{157, 158, 159, 160, 161, 162, 163, 164, 0, 0, 0, 0, 165, 166, 167, 168, 169, 170, 171, 0},\
{172, 173, 174, 175, 176, 177, 178, 179, 0, 0, 0, 0, 0, 180, 181, 182, 183, 184, 185, 0},\
{186, 187, 188, 189, 190, 191, 192, 193, 194, 0, 0, 0, 0, 195, 196, 197, 198, 199, 200, 0},\
{201, 202, 203, 204, 205, 206, 207, 208, 209, 0, 0, 0, 0, 210, 211, 212, 213, 214, 0, 0},\
{215, 216, 217, 218, 219, 220, 221, 222, 223, 0, 0, 0, 0, 224, 225, 226, 227, 0, 0, 0},\
{228, 229, 230, 231, 232, 233, 234, 235, 236, 0, 0, 0, 0, 0, 237, 238, 0, 0, 0, 0},\
{239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 279, 280, 281, 282, 283, 284, 285, 286, 287, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 0, 288, 289, 290, 291, 292, 293, 294, 295, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 0, 296, 297, 298, 299, 300, 301, 302, 303, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
{0, 0, 0, 0, 304, 305, 306, 307, 308, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},\
}
```

right palm

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 0, 0, 0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0, 0, 0, 0, 5, 6, 7, 8, 9, 10, 11, 12, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0, 0, 0, 0, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 0, 0}, \
{0, 0, 0, 0, 0, 0, 0, 0, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 0}, \
{0, 0, 0, 0, 0, 0, 0, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 0}, \
{0, 0, 0, 0, 0, 0, 0, 0, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 58, 59, 60, 61, 62}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 63, 64}, \
{0, 0, 0, 0, 0, 65, 66, 67, 68, 69, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, \
{0, 0, 70, 71, 72, 73, 74, 75, 76, 77, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, \
{0, 78, 79, 80, 81, 82, 83, 84, 85, 86, 0, 0, 0, 0, 0, 0, 0, 87, 88, 0}, \
{0, 89, 90, 91, 92, 93, 94, 95, 96, 0, 0, 0, 0, 0, 0, 0, 97, 98, 99, 0}, \
{100, 101, 102, 103, 104, 105, 106, 107, 108, 0, 0, 0, 0, 0, 0, 109, 110, 111, 112, 113}, \
{114, 115, 116, 117, 118, 119, 120, 121, 122, 0, 0, 0, 0, 0, 123, 124, 125, 126, 127, 128}, \
{129, 130, 131, 132, 133, 134, 135, 136, 0, 0, 0, 0, 0, 137, 138, 139, 140, 141, 142, 0}, \
{143, 144, 145, 146, 147, 148, 149, 150, 0, 0, 0, 0, 0, 151, 152, 153, 154, 155, 156, 0}, \
{0, 157, 158, 159, 160, 161, 162, 163, 0, 0, 0, 0, 164, 165, 166, 167, 168, 169, 170, 171}, \
{0, 172, 173, 174, 175, 176, 177, 0, 0, 0, 0, 0, 178, 179, 180, 181, 182, 183, 184, 185}, \
{0, 186, 187, 188, 189, 190, 191, 0, 0, 0, 0, 192, 193, 194, 195, 196, 197, 198, 199, 200}, \
{0, 0, 201, 202, 203, 204, 205, 0, 0, 0, 0, 206, 207, 208, 209, 210, 211, 212, 213, 214}, \
{0, 0, 0, 215, 216, 217, 218, 0, 0, 0, 0, 219, 220, 221, 222, 223, 224, 225, 226, 227}, \
{0, 0, 0, 0, 228, 229, 0, 0, 0, 0, 0, 230, 231, 232, 233, 234, 235, 236, 237, 238}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 279, 280, 281, 282, 283, 284, 285, 286, 287, 0}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 288, 289, 290, 291, 292, 293, 294, 295, 0, 0}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 296, 297, 298, 299, 300, 301, 302, 303, 0, 0}, \
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 304, 305, 306, 307, 308, 0, 0, 0, 0}  \
}
```

2.3 Global Indices and Region Slices for the 746 Points

The layouts above show the sparse matrix for each finger. When the total tactile length T is 746, the SDK divides the tactile array into 15 regions using the fixed, contiguous slices below. A half-open interval [a, b) covers global indices a through b − 1. Each region's own data array is indexed from 0. The two tables differ in granularity but describe the same sensing points. **For example, the thumb's 66 points are 38 tip points plus 28 pad points.**

Table 2

Region	Points	Global Slice	Matrix Bounds
thumb.tip	38	[0,38)	8×7
thumb.pad	28	[38,66)	4×7
index.tip	38	[66,104)	8×7
index.pad	28	[104,132)	4×7
index.root	28	[132,160)	4×7
middle.tip	37	[160,197)	8×7
middle.pad	28	[197,225)	4×7
middle.root	28	[225,253)	4×7
ring.tip	37	[253,290)	8×7
ring.pad	28	[290,318)	4×7
ring.root	28	[318,346)	4×7
pinky.tip	39	[346,385)	8×7
pinky.middle	28	[385,413)	4×7
pinky.root	25	[413,438)	4×7
palm.surface	308	[438,746)	30×20

Rows and columns give the bounding box of the sparse physical matrix. **Do not assume rows × columns = count.In a layout matrix, 0 means no sensor**. A positive integer k is the 1-based local index in the region's data array, so read it as data[k − 1].

An empty matrix cell is not the same as an active sensor that reads 0.

For example, local index k in `index.tip` maps to global flat index `66 + k − 1`. In **POSE mode 1**, the spatial coordinate points[66 + k − 1] pairs with the same sensing point, with XYZ in meters.

- **Always follow the fixed point order. Do not number EVERY matrix cell CONSECUTIVELY in its place.**

Each sensing point is a uint8 value `0~255` at 1 g/LSB: a value of **n** means **n** g, so each point covers 0–255 g. g is a gram scale for tactile load. **Do not read it as acceleration (g), and do not treat raw values as newtons.** The two figures below show the sensing point layout for the left and right hands (yellow numbers are local indices within each region). To check the matrix mapping, see Chapter 4 of the **OmniGlove SDK User Guide.**



Figure 2



Figure 3

2.4 Tactile Point Cloud

Each frame gives the position and force of all 746 sensing points on the glove. Positions are the 3D positions of the sensing points in the wrist frame and move with the hand. They are derived from the poses of the 21 landmarks solved by IK.

3 POSE: Pose Measurement Semantics

POSE provides six named nodes: thumb, index, middle, ring, pinky, and wrist. Each node contains EMF and IMU data. EMF gives the device's native position and orientation. For each node, check `emf.valid` and `confidence_milli` before you read position and orientation. Use the native units or convert as specified for each field, and keep the validity and quality values with the data.

Table 3

EMF Field	Type and Unit	How to Read
position_mm[3]	int16；x/y/z，mm	Multiply by 0.001 to convert to meters.
yaw_pitch_roll_cdeg[3]	int16；yaw/pitch/roll，0.01°	Multiply by 0.01 to convert to degrees. For rotation conventions, see **Coordinate Frames and Transformations**.
confidence_milli	uint16；0～1000	Keep the quality value and use it together with valid.
valid	uint8 validity flag	Check it before you use the measurement.

POSE uses millimeters and Euler angles, while **POST** uses meters and quaternions. **Do not mix the two formats.** To access and check the DIRECT structures (`omniglove_pose_node_t` / `omniglove_pose_emf_t`) and the emf object of UDP nodes, see Chapter 5 of the **OmniGlove SDK User Guide**.

4 POST: Joint and Landmark Semantics

POST is the hand model output from on-device post-processing and IK. It uses the named objects thumb, index, middle, ring, pinky, and wrist. The 21 joints are ordered as 5 thumb joints followed by 4 joints for each of the other fingers. The 21 landmarks are ordered as Wrist, 4 thumb points, then 4 points each for the index, middle, ring, and little fingers.

Table 4

Object	Field Order	Unit/Reference
Thumb joints	cmc_rotation、cmc_abduction、cmc_flexion、mcp_flexion、ip_flexion	rad
Finger joints	mcp_abduction、mcp_flexion、pip_flexion、dip_flexion	rad
Thumb landmarks	cmc、mcp、ip、tip	Wrist frame；m；xyzw
Finger landmarks	mcp、pip、dip、tip	Wrist frame；m；xyzw

Object	Field Order	Unit/Reference
rx_tip_target	One per finger, thumb to little finger	Wrist frame； m； xyzw
wrist.landmark	Wrist reference marker	Identity pose; not a measurement of the external wrist position
wrist.quaternion_xyzw	Wrist orientation	Relative to the calibration zero; read according to validity

Landmarks are model joints or fingertip keypoints. `rx_tip_target` is a fingertip target derived from the sensors. The two have different meanings and are not interchangeable. The system does not output a global wrist position in room, camera, or robot base coordinates.

4.1 Result Fields and Flags

Table 5

Field	Meaning	Checks and Use
joints_rad	5 thumb joints and 4 joints for each other finger	rad. For sign conventions, see **Coordinate Frames and Transformations**.
landmarks / wrist.landmark	position_m[3]、quaternion_xyzw[4]	Position: float32, m. Quaternion: float32, xyzw.
rx_tip_target	One sensor-derived target per finger	Interpret separately from the model landmarks.
wrist.quaternion_xyzw	float32[4] wrist orientation	Check the wrist-orientation-valid bit.
device_tick_ms / frame_id	Device tick / uint32 POST frame number	Interpret separately from the shared device_sequence.
flags bit0	IK converged	Filter according to your task's quality requirements.
flags bit1	Wrist orientation valid	Determines whether wrist orientation can be used.
flags bit2	Skeleton model valid	Check personal calibration and deployment.

The SDK does not discard a decoded POST frame because a flag bit is not set. Your application should check the fields its task needs. Undefined bits are passed through

unchanged. For fields and the check procedure, see Sections 3.1–3.2 of the **OmniGlove SDK User Guide**.

5 IMU Semantics

Public IMU data consists of Euler angles and status flags within POSE. Each of the six named nodes (five fingers and wrist) contains imu, and there is also a back-of-hand IMU (`back_of_hand_imu` in DIRECT, `back_of_hand.imu` in UDP). **IMU components are ordered roll, pitch, yaw, which differs from the EMF order in `yaw_pitch_roll_cdeg`**. Convert by named field. **Do not reuse one array order for both.**

Table 6

Field	Type or Bit	Meaning
roll_pitch_yaw_cdeg[3]	int16；0.01°	Ordered roll、pitch、yaw；Multiply by 0.01 to convert to degrees.
flags bit0	Sample flag	An IMU sample is present in this cycle.
flags bit1	yaw lock flag	yaw is locked
reserved	uint8	Reserved

To read IMU data, check flags `bit0` for each node separately. If it is not set, do not treat that node as a new measurement for this cycle. Before you use yaw, check `bit1` and keep its lock state. Keep the roll/pitch/yaw order when you convert angles. Tactile-only configurations do not read the six nodes or the back-of-hand IMU. For the reading procedure and timing, see Chapter 6 and Section 3.2.2 of the **OmniGlove SDK User Guide**.