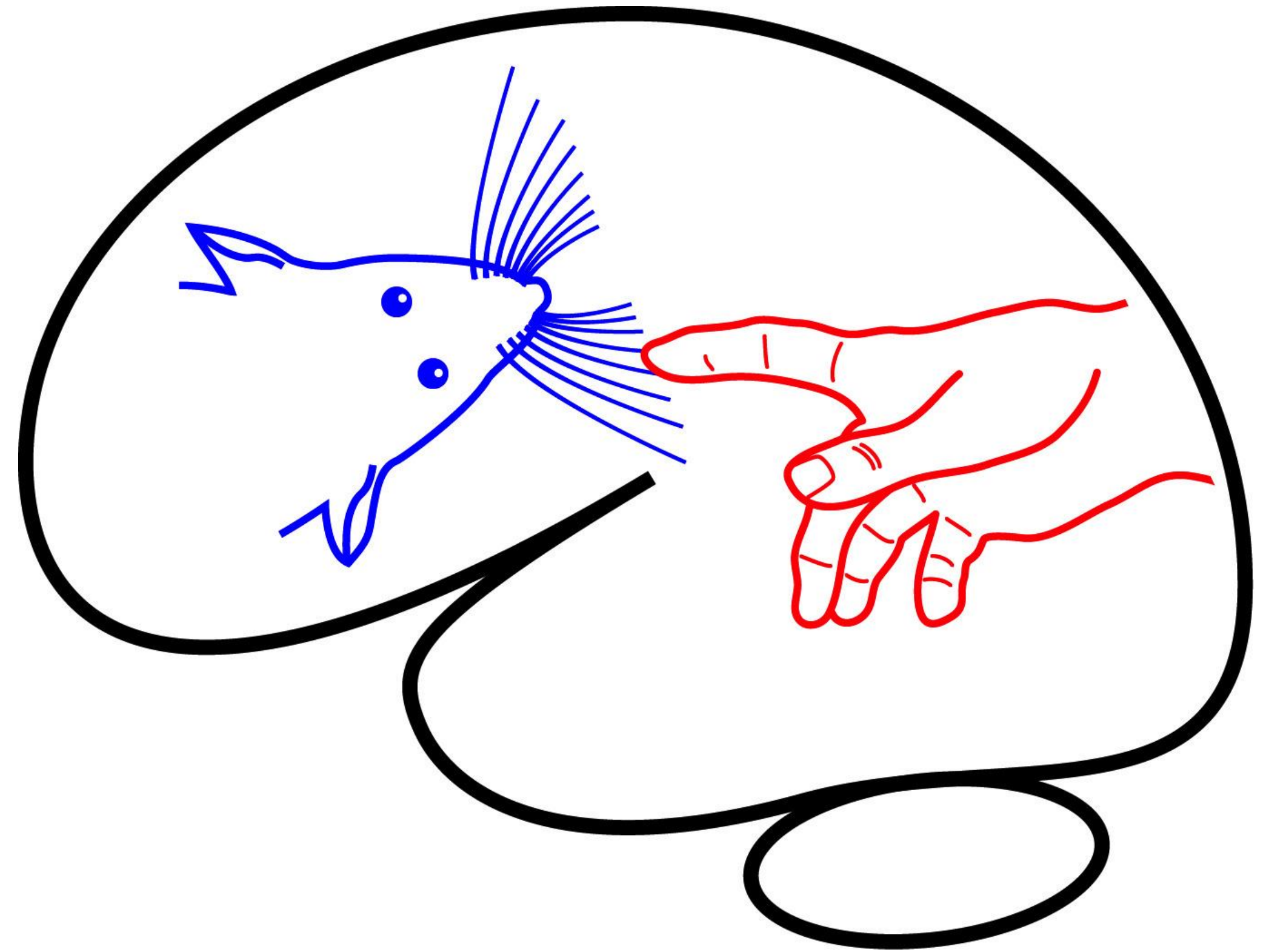


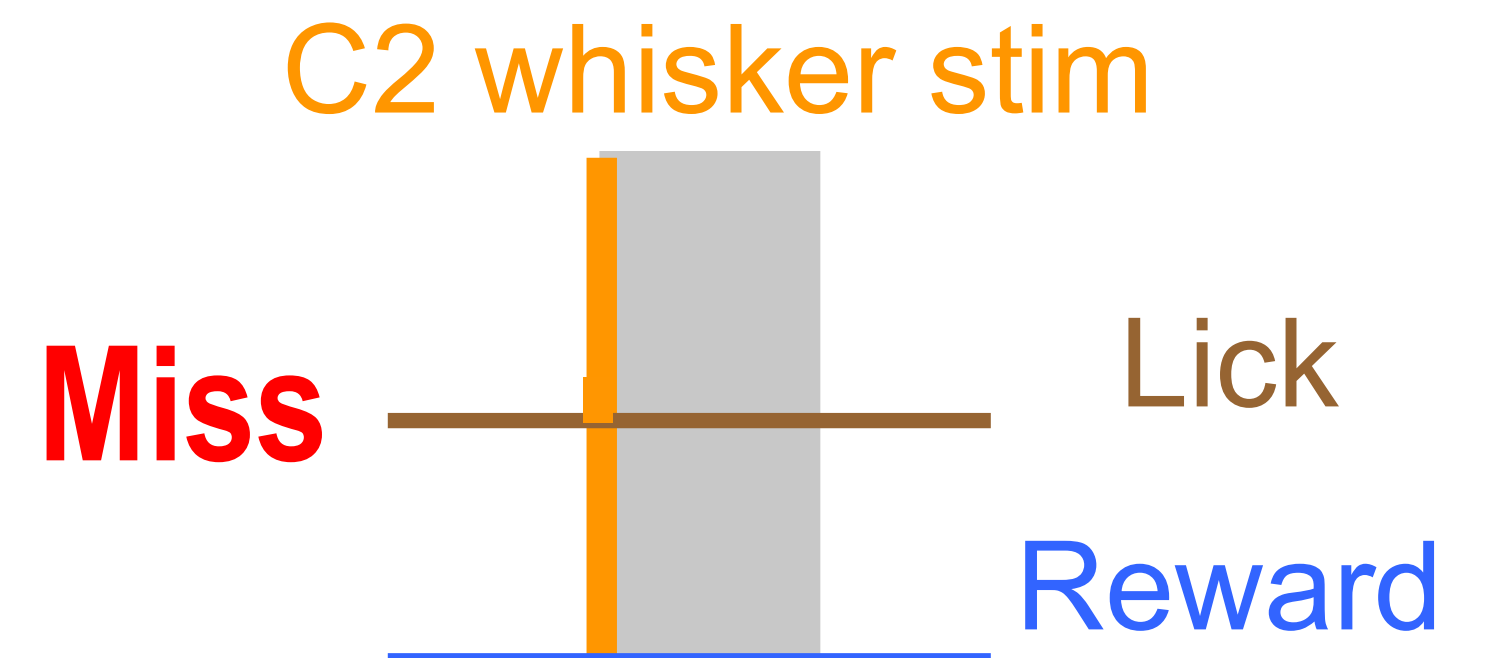
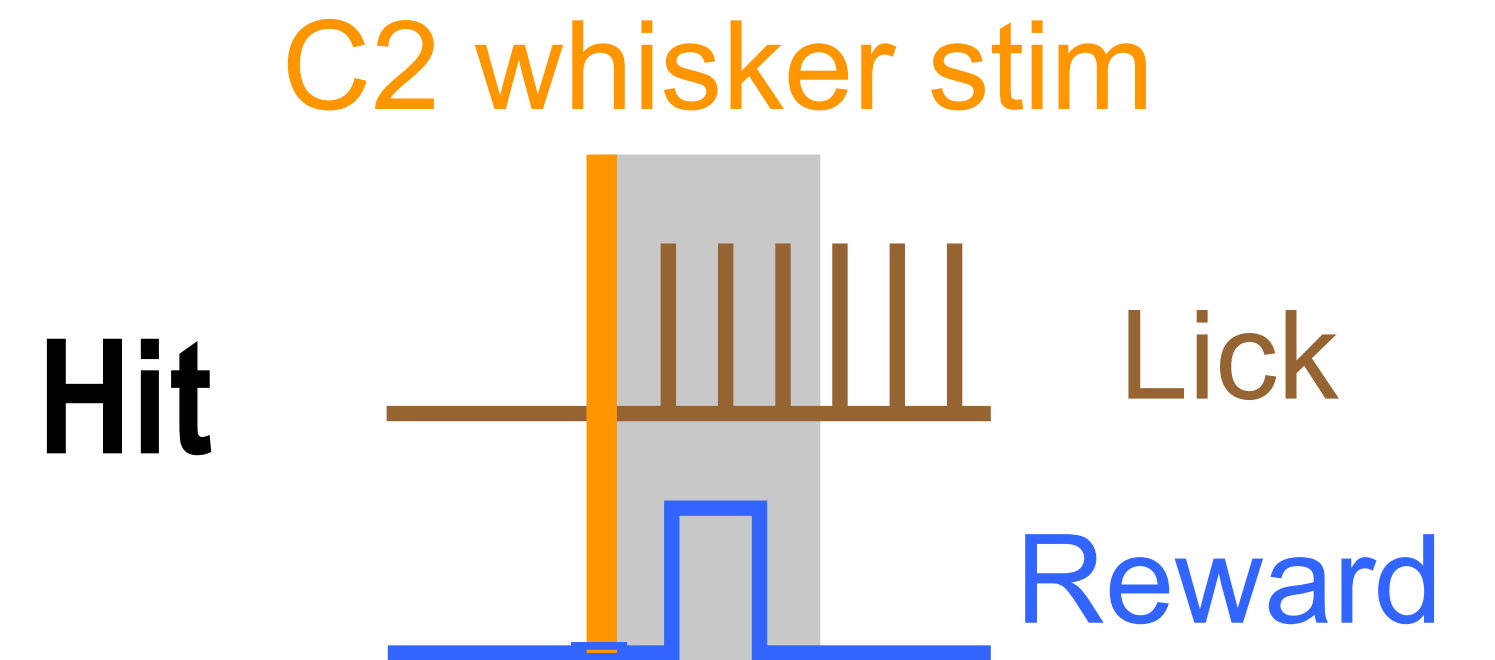
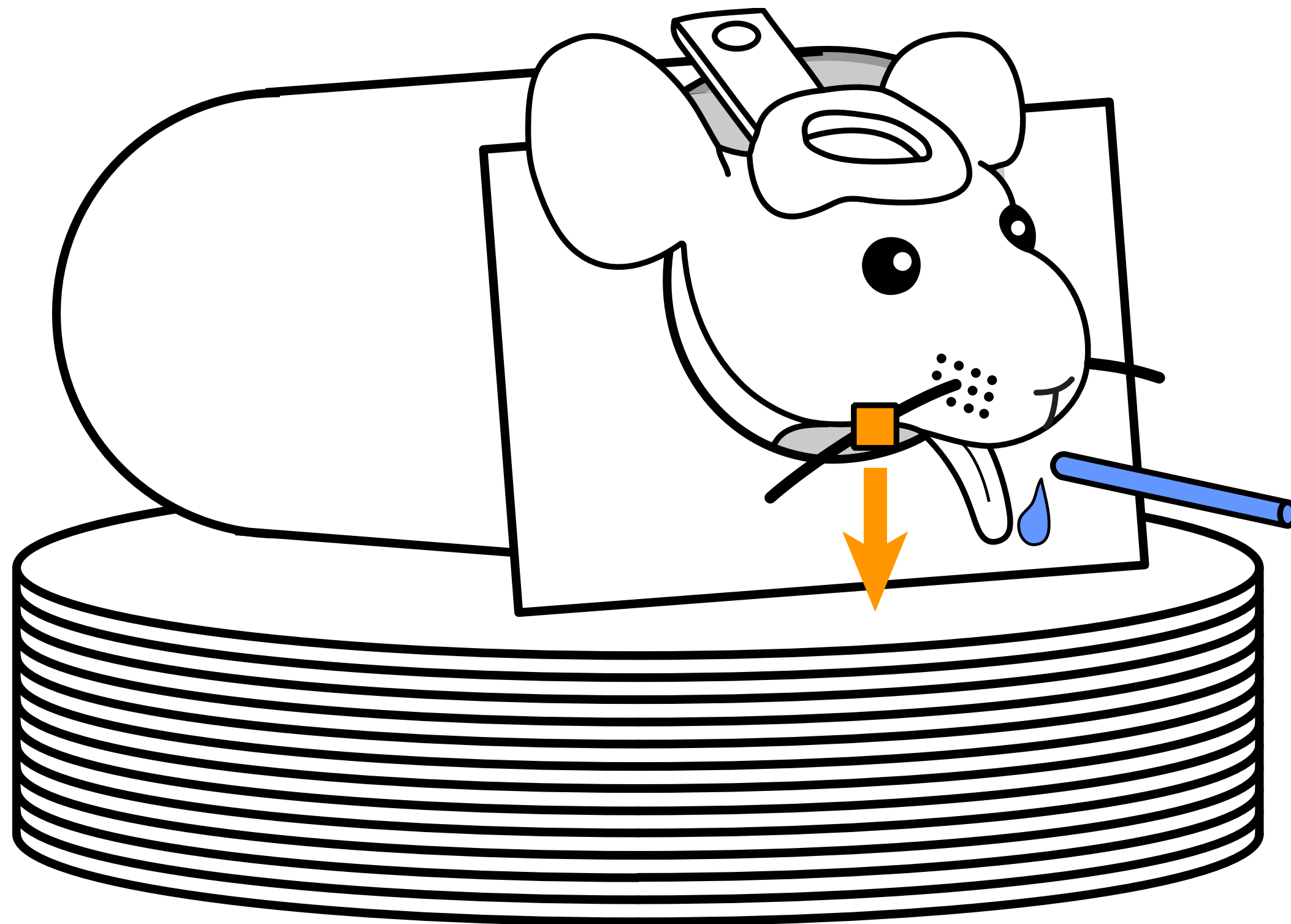
EPFL



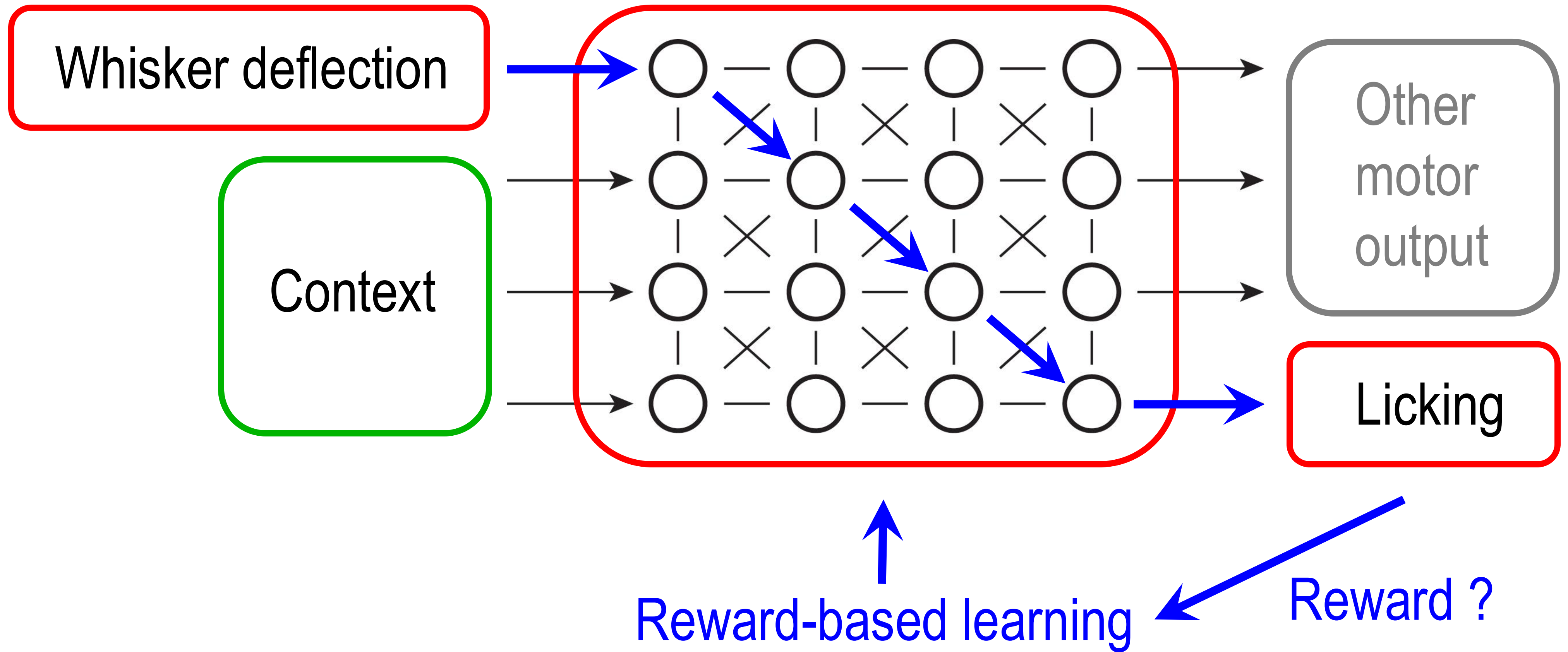
Context-dependent sensorimotor transformation

Carl Petersen

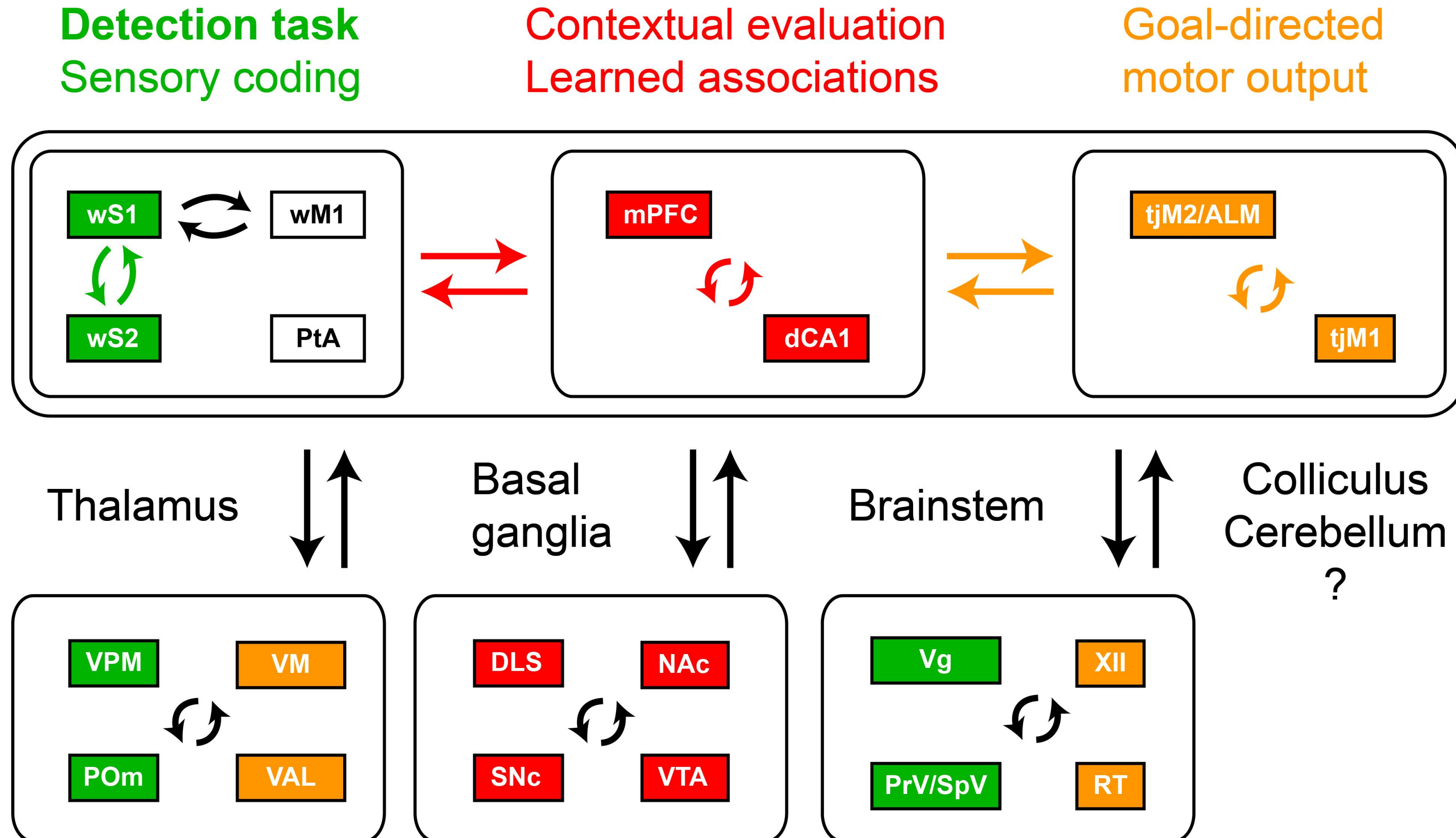
Whisker detection task



Neural circuits for sensorimotor transformation

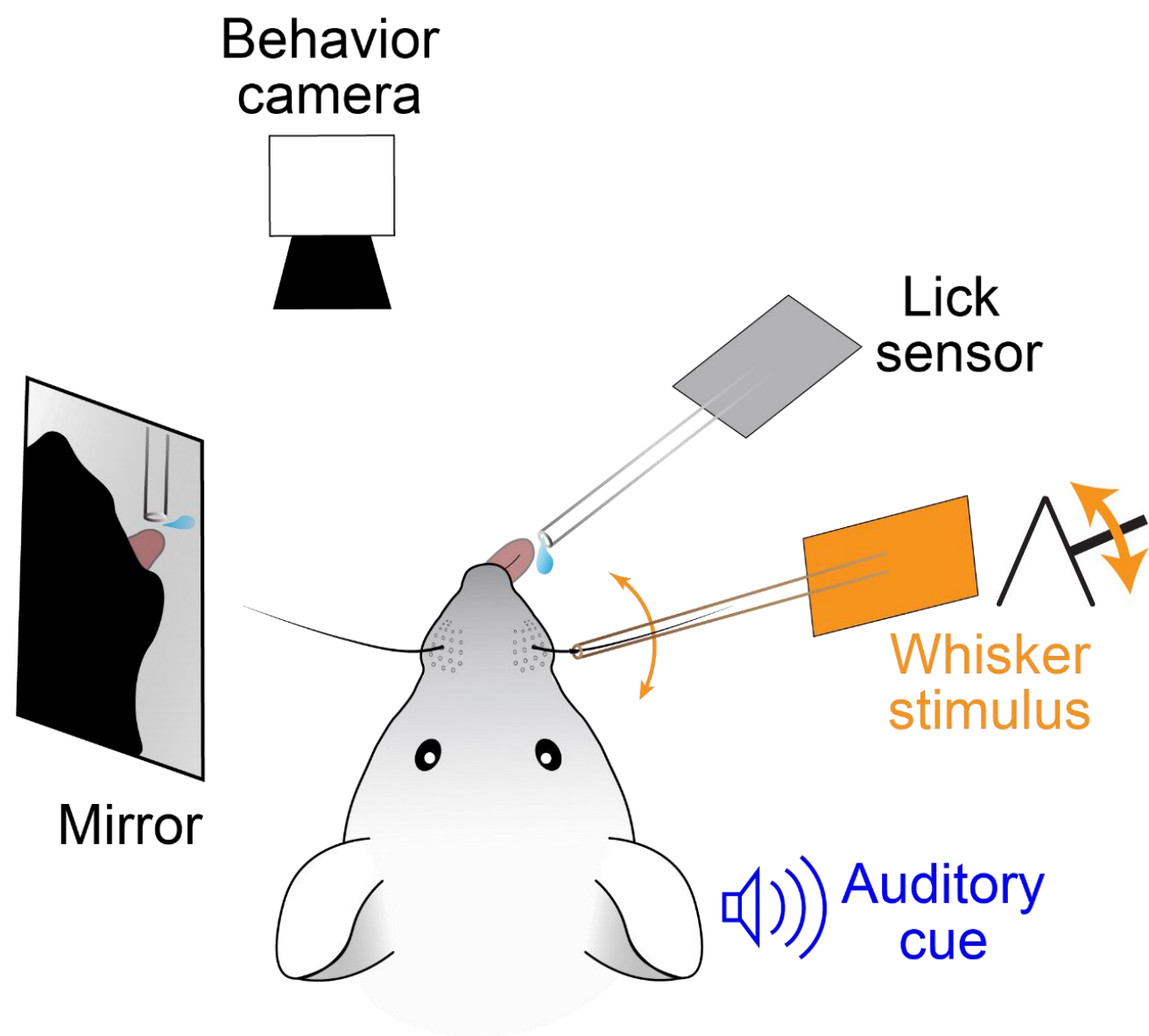


Complex brain-wide neuronal circuits

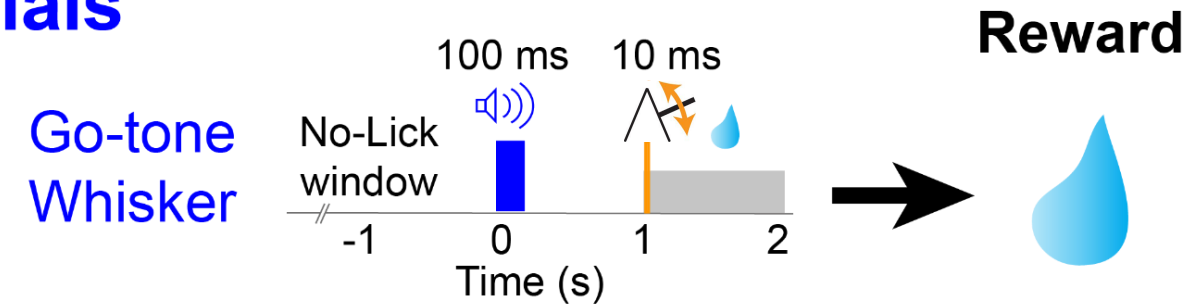




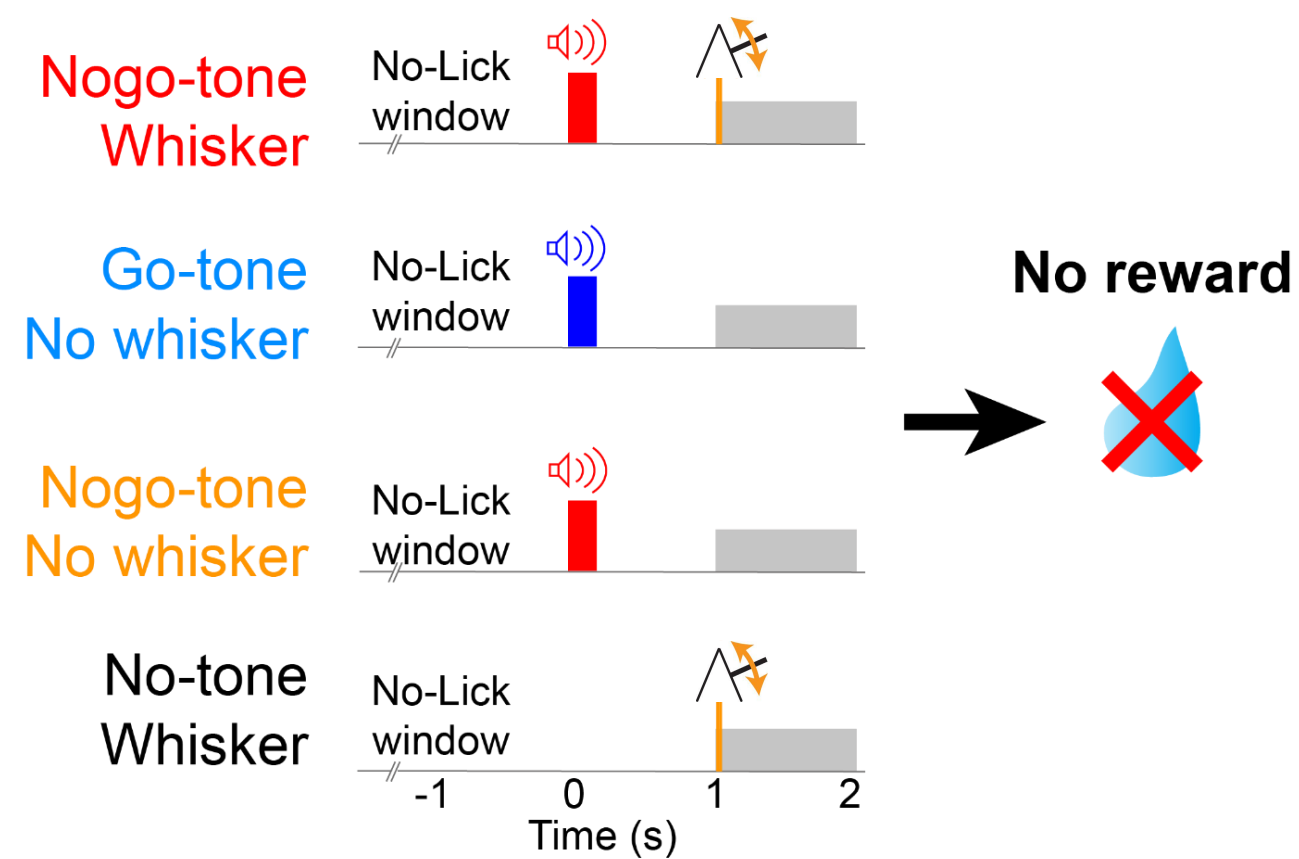
Context-dependent whisker detection task



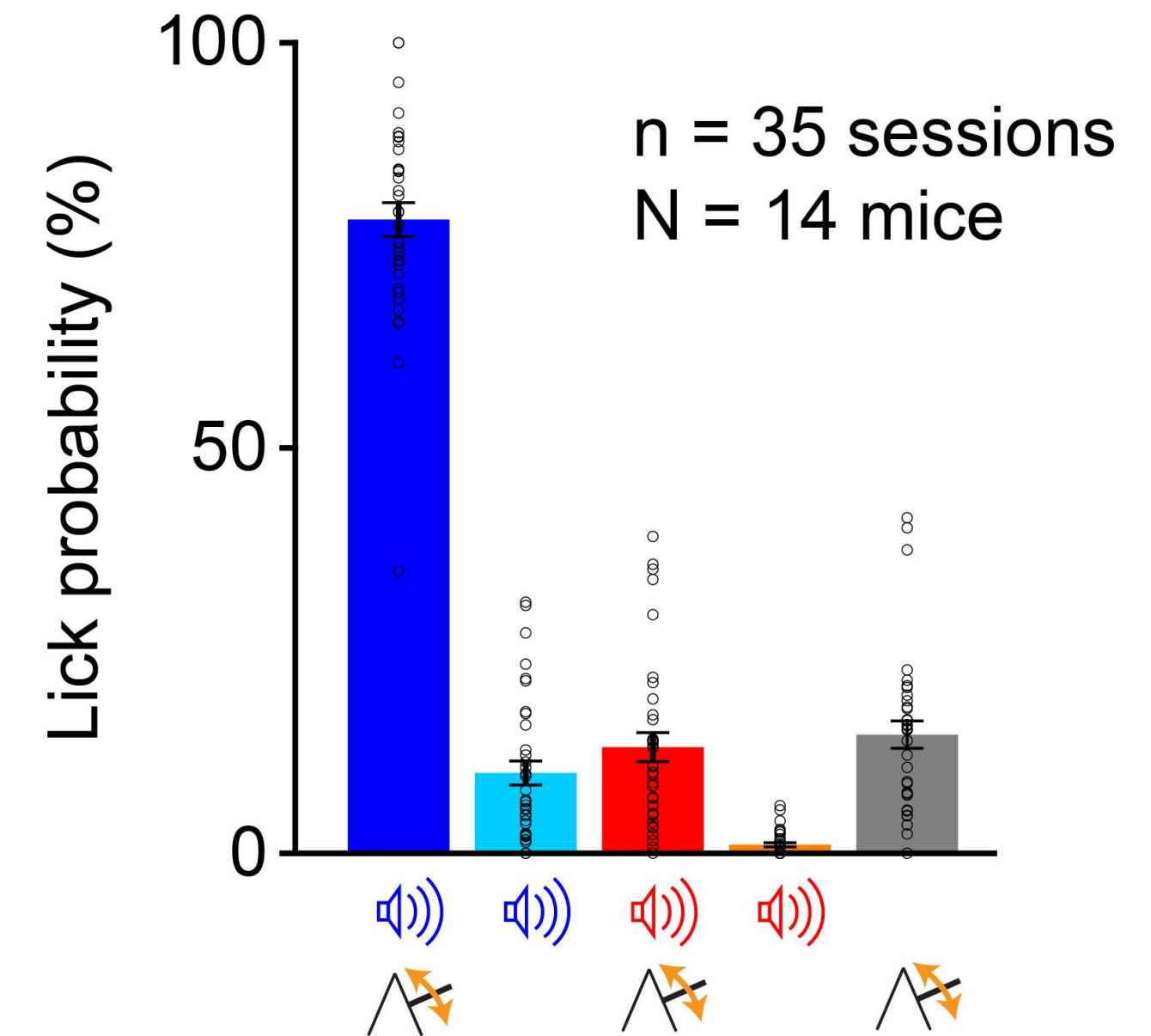
Go trials



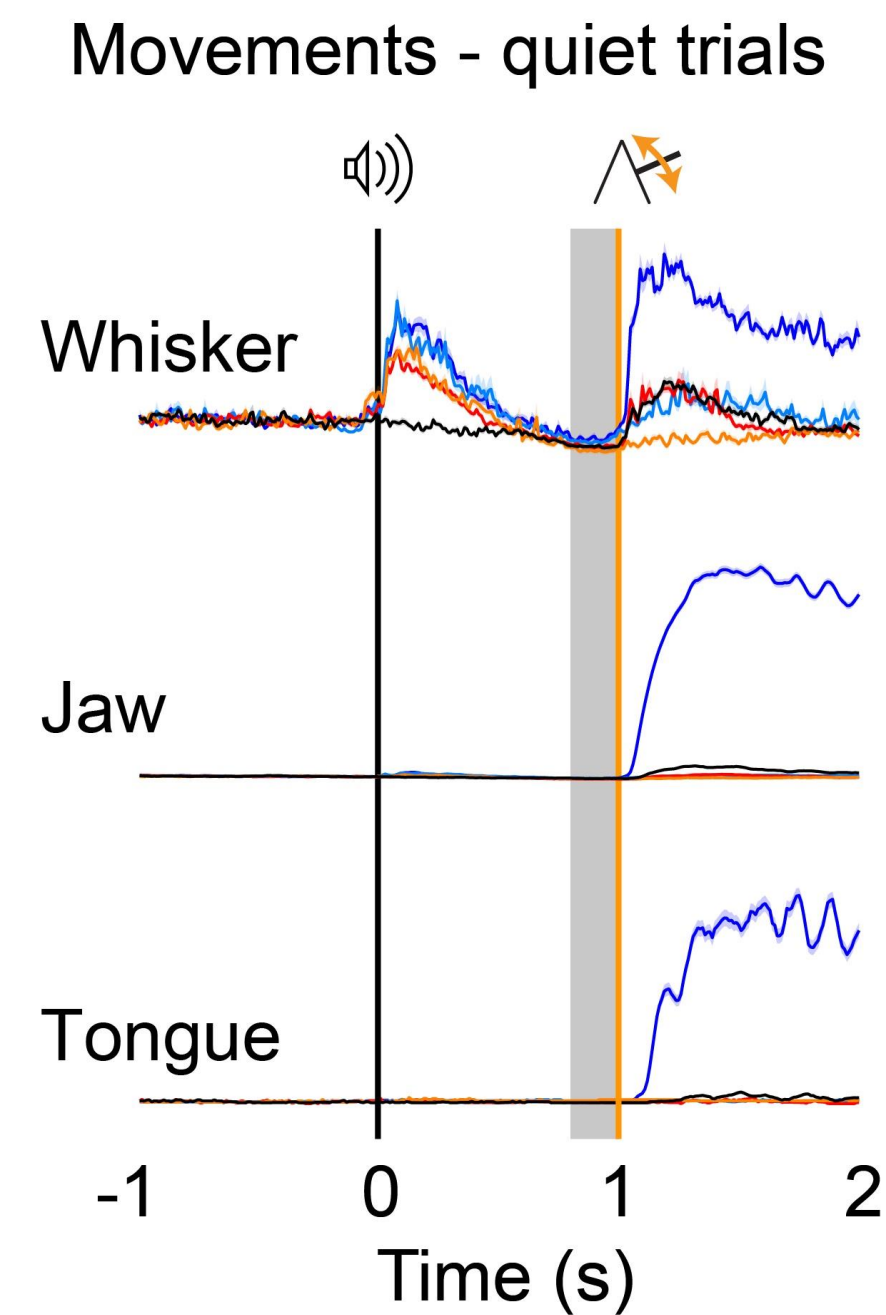
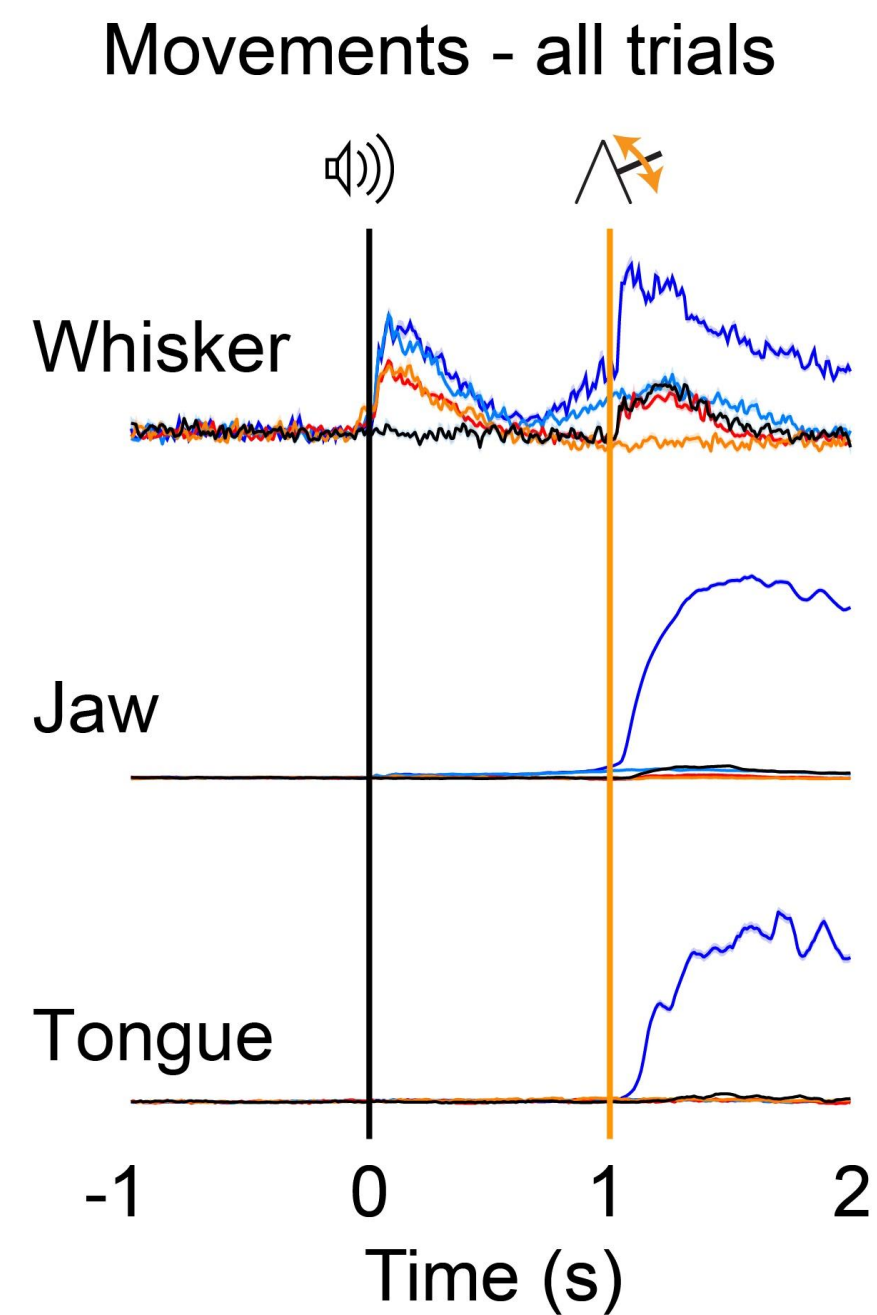
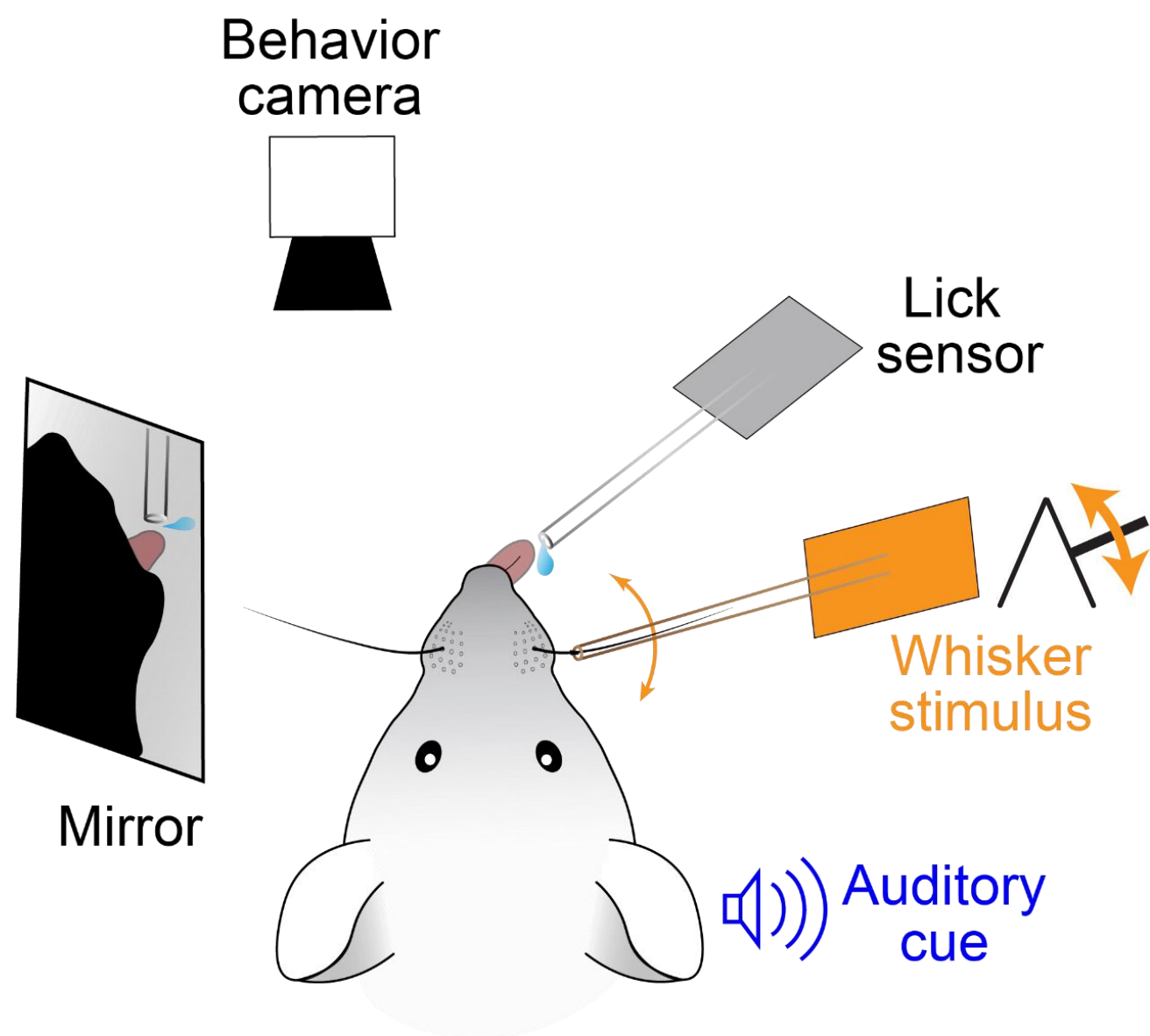
Nogo trials



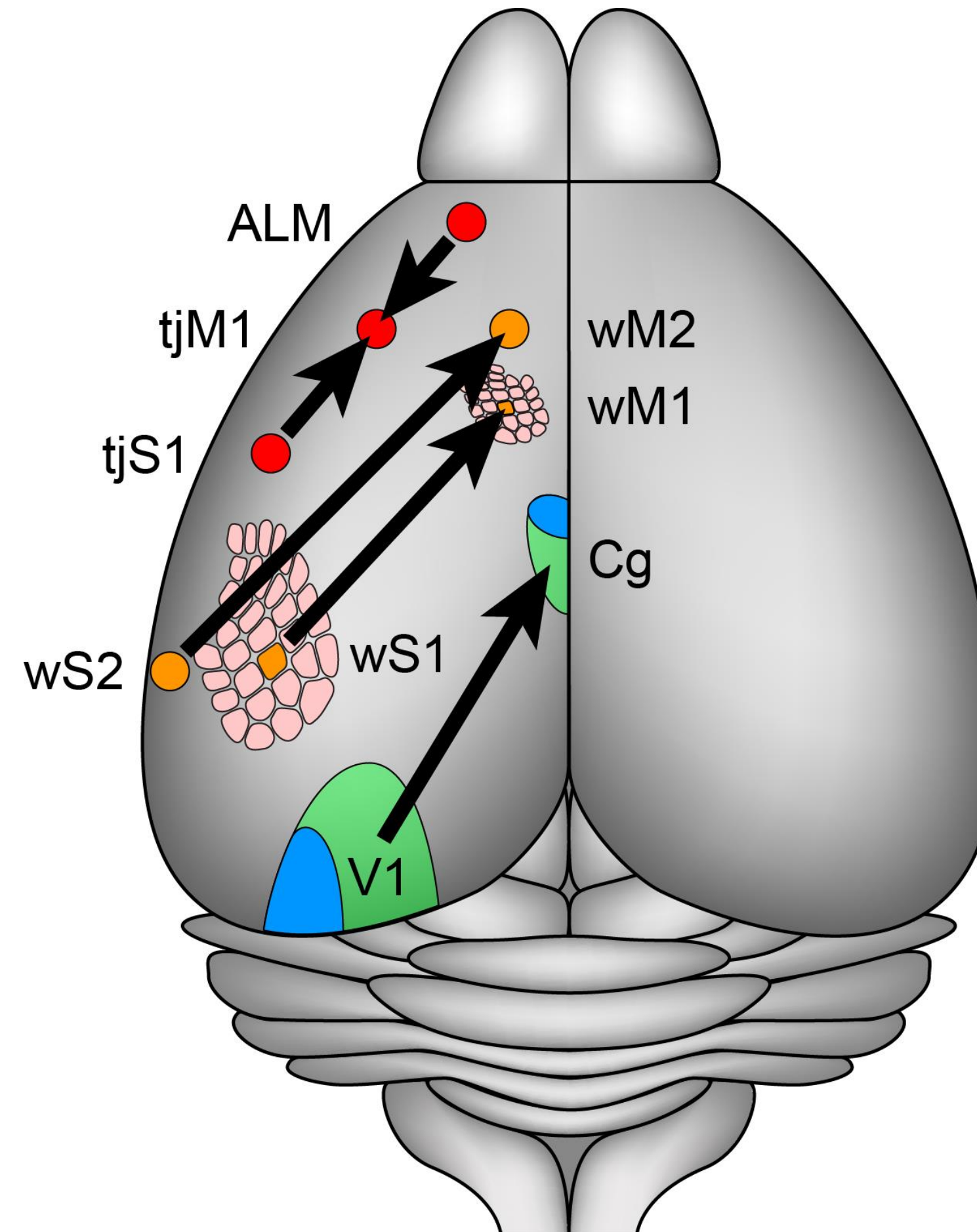
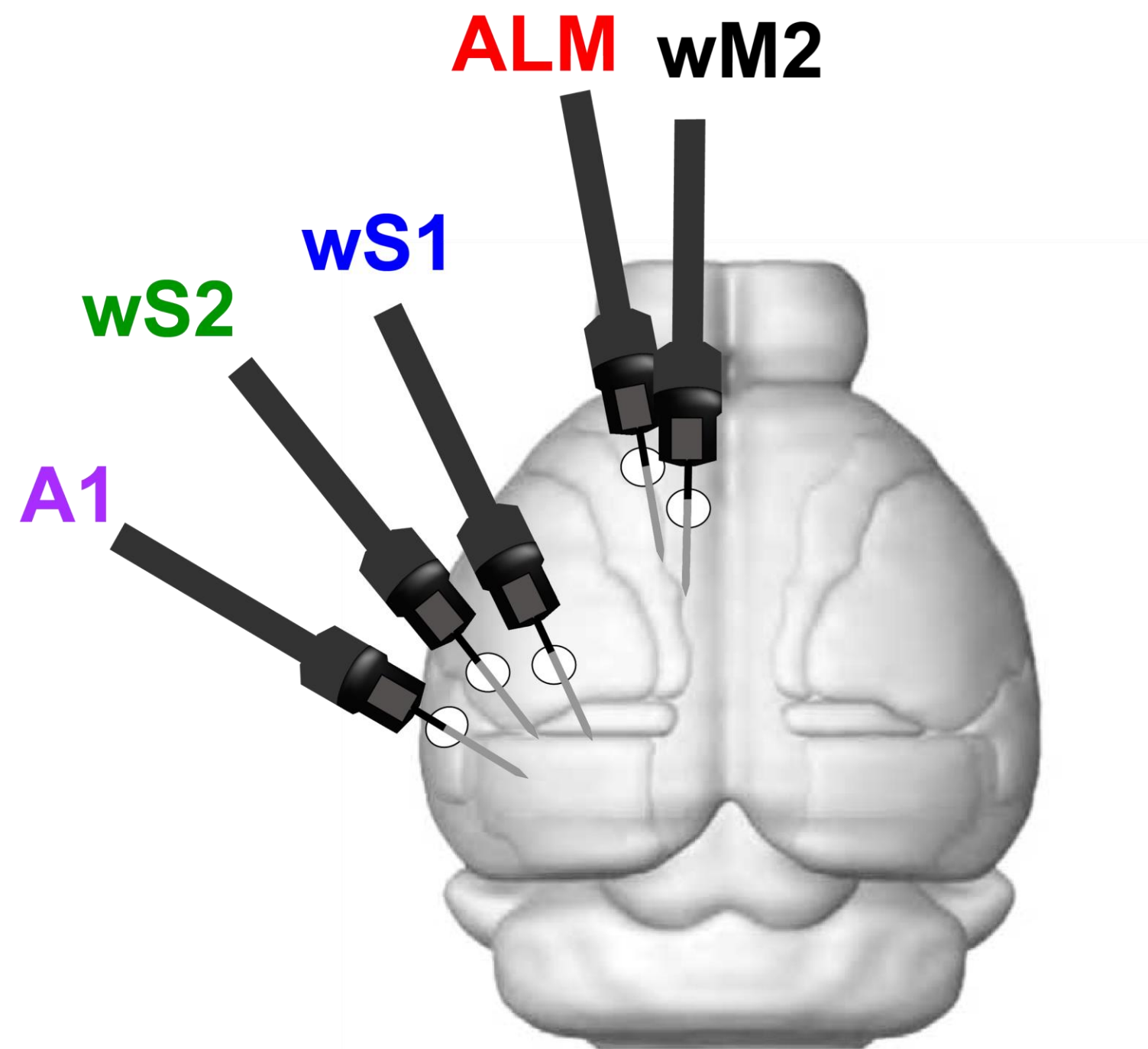
During electrophysiology



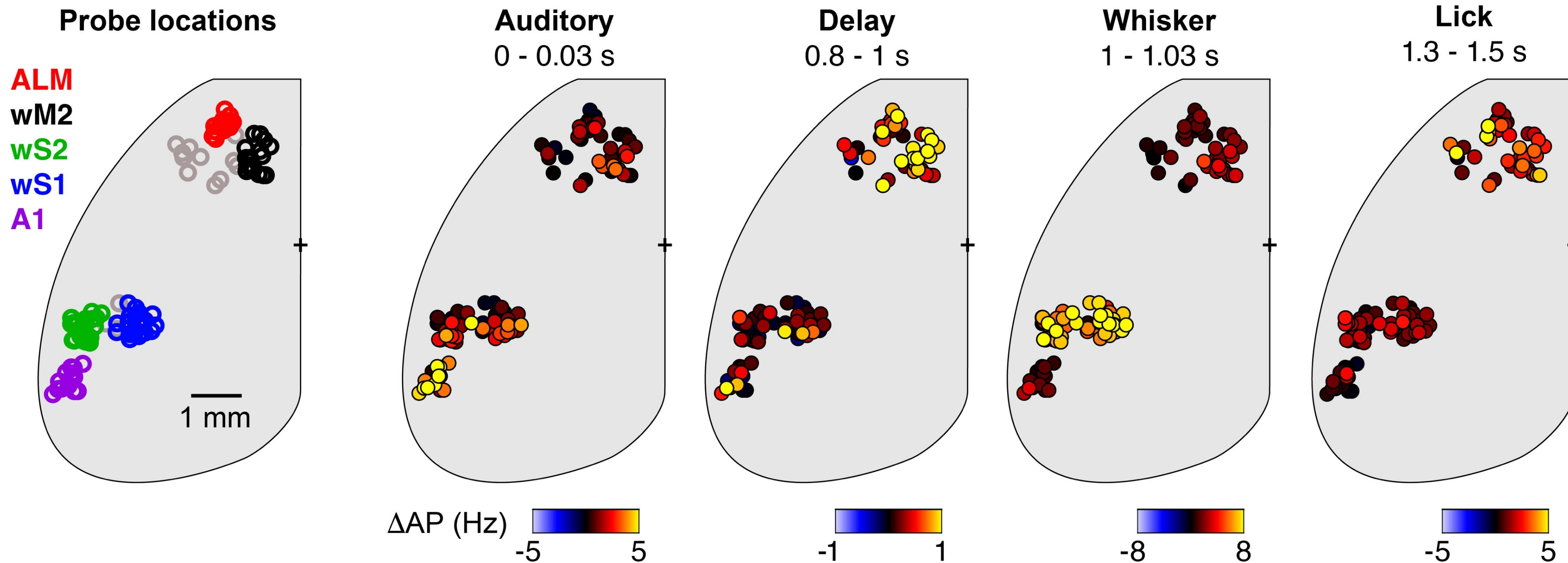
Orofacial movements



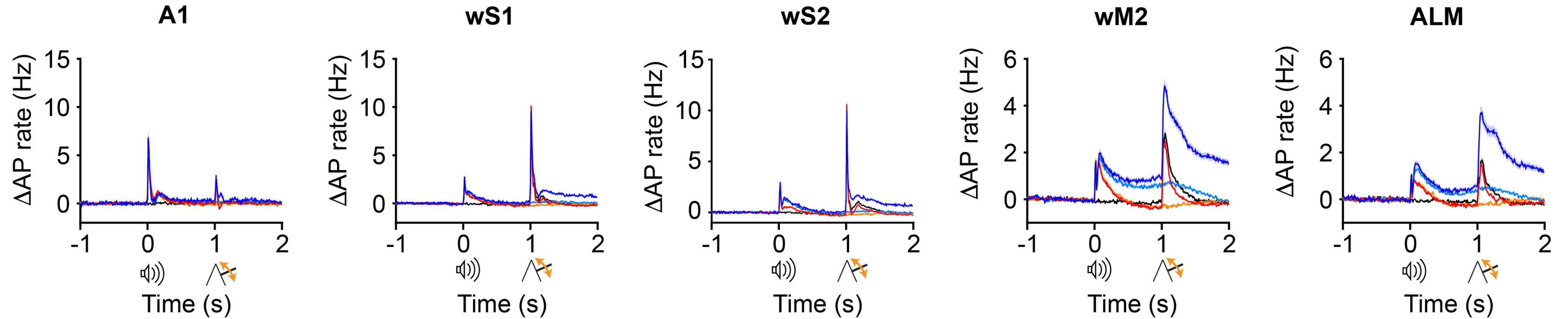
Targeted electrophysiological recordings



Cortical activity maps



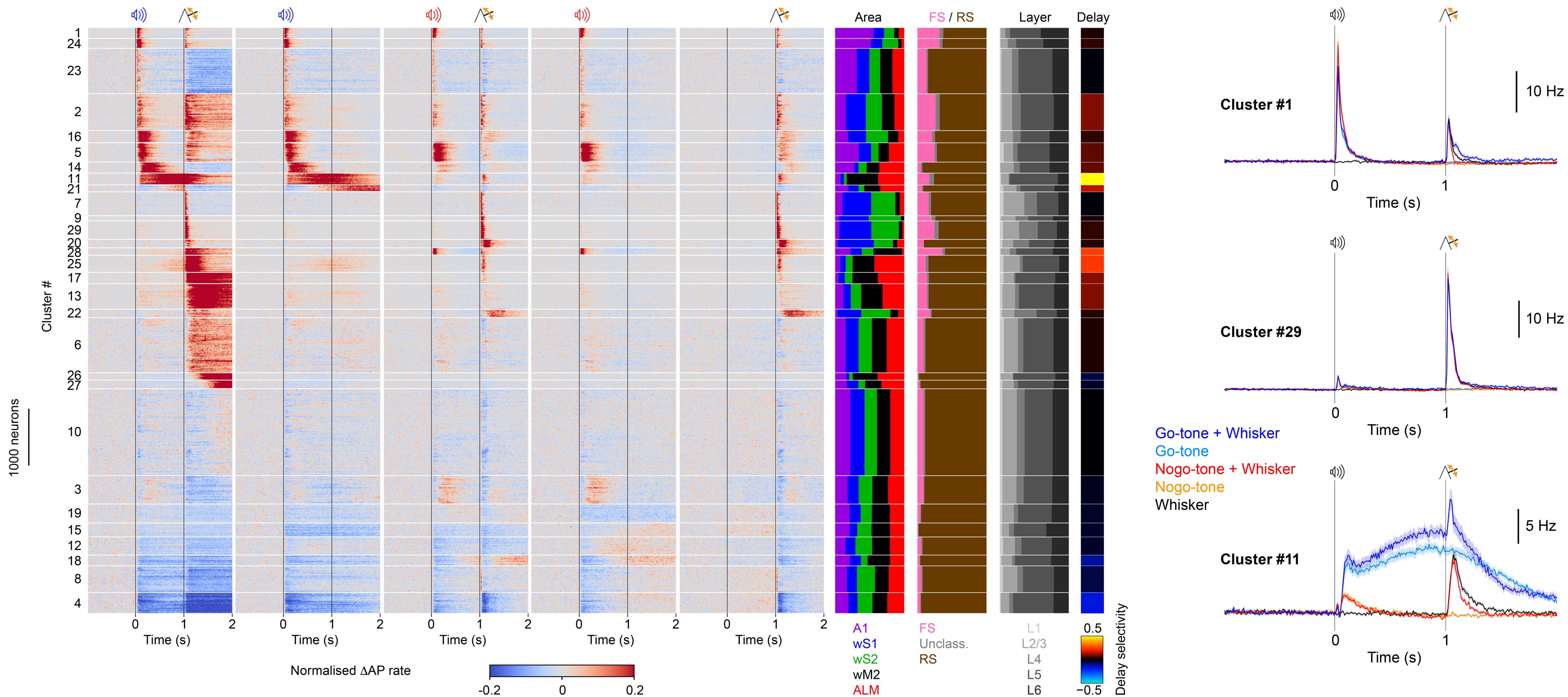
PSTHs for correct trials



Correct trial PSTHs

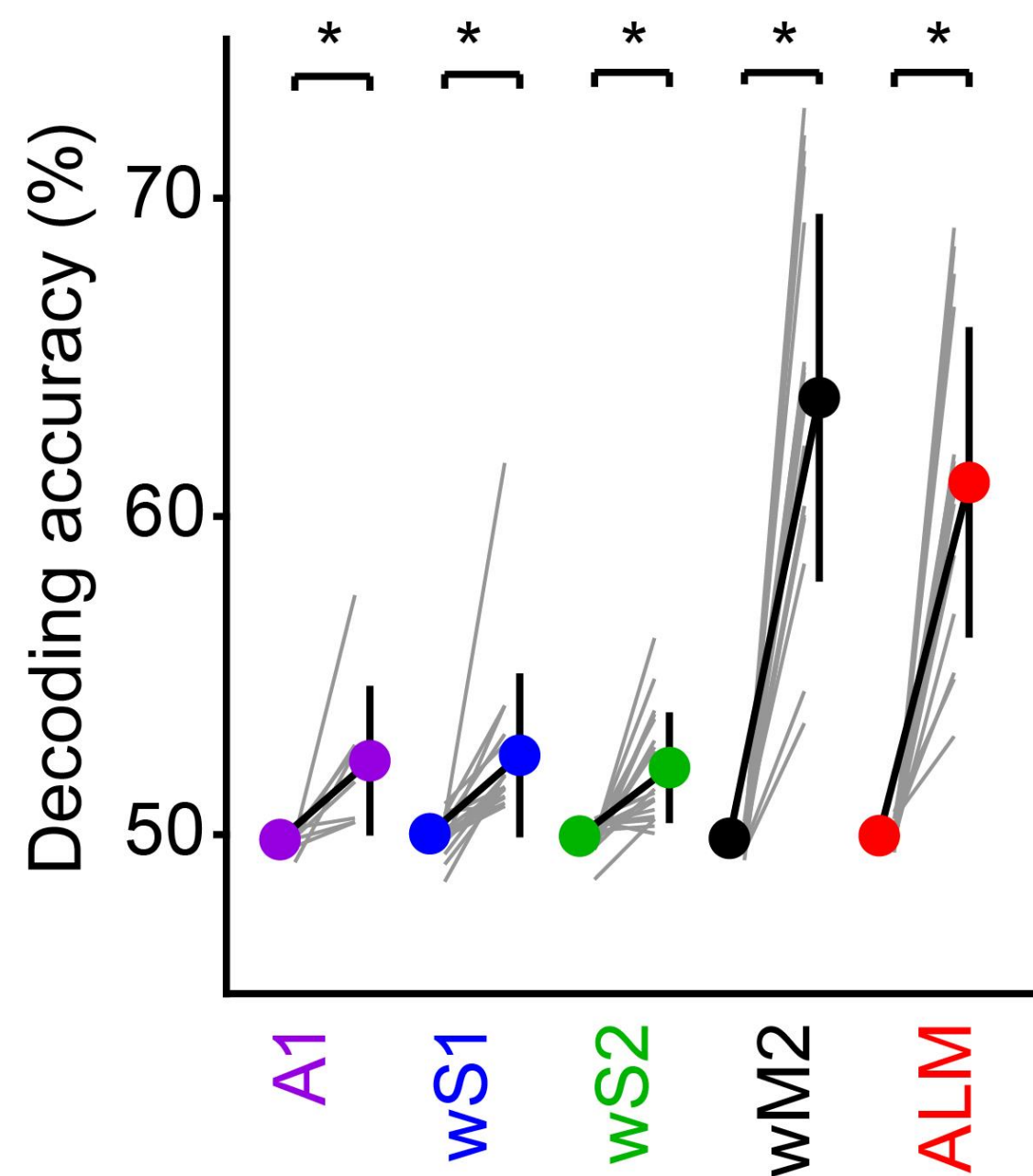
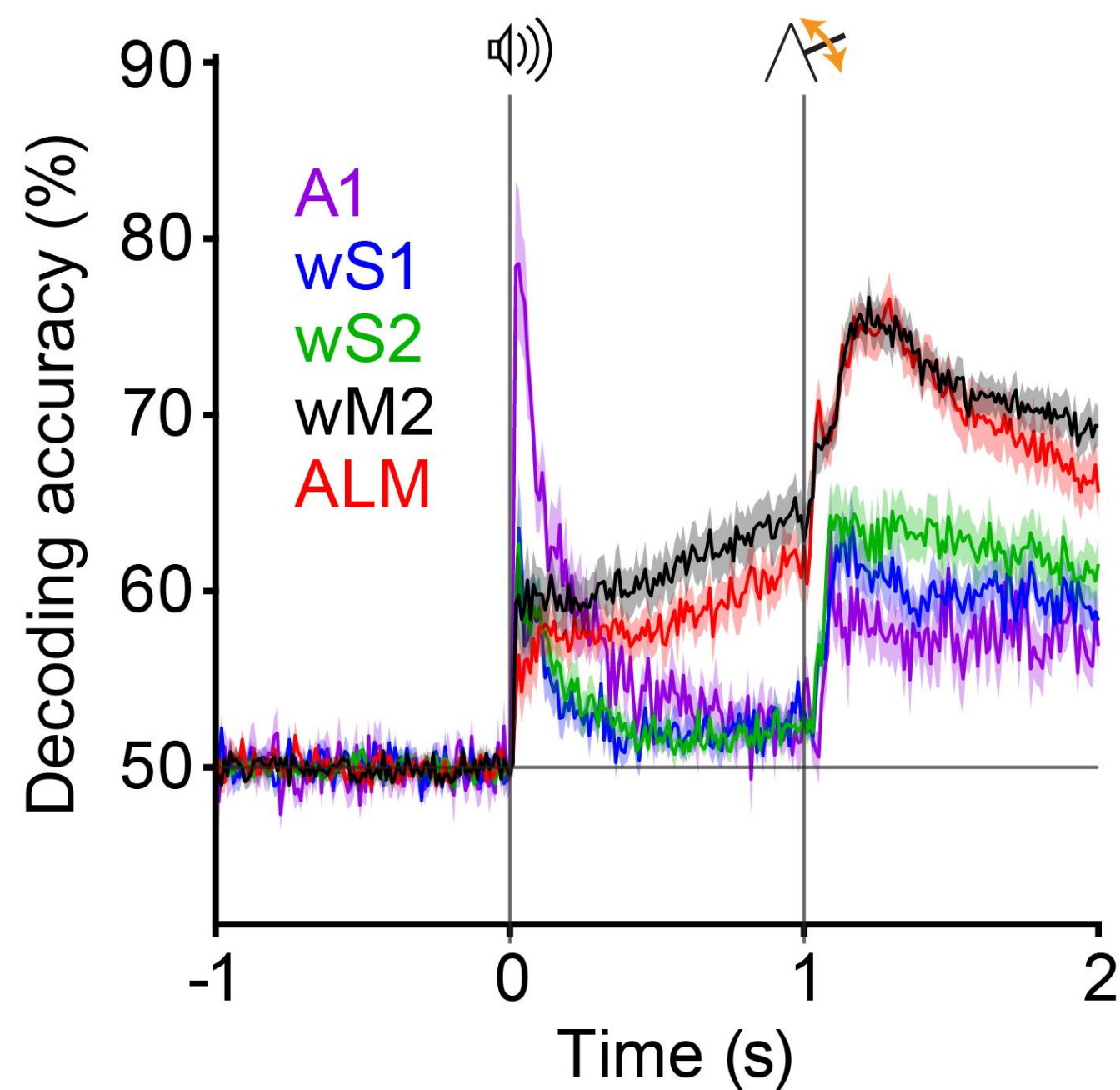
Go-tone Whisker
Go-tone
Nogo-tone Whisker
Nogo-tone
Whisker

Clustering of firing dynamics

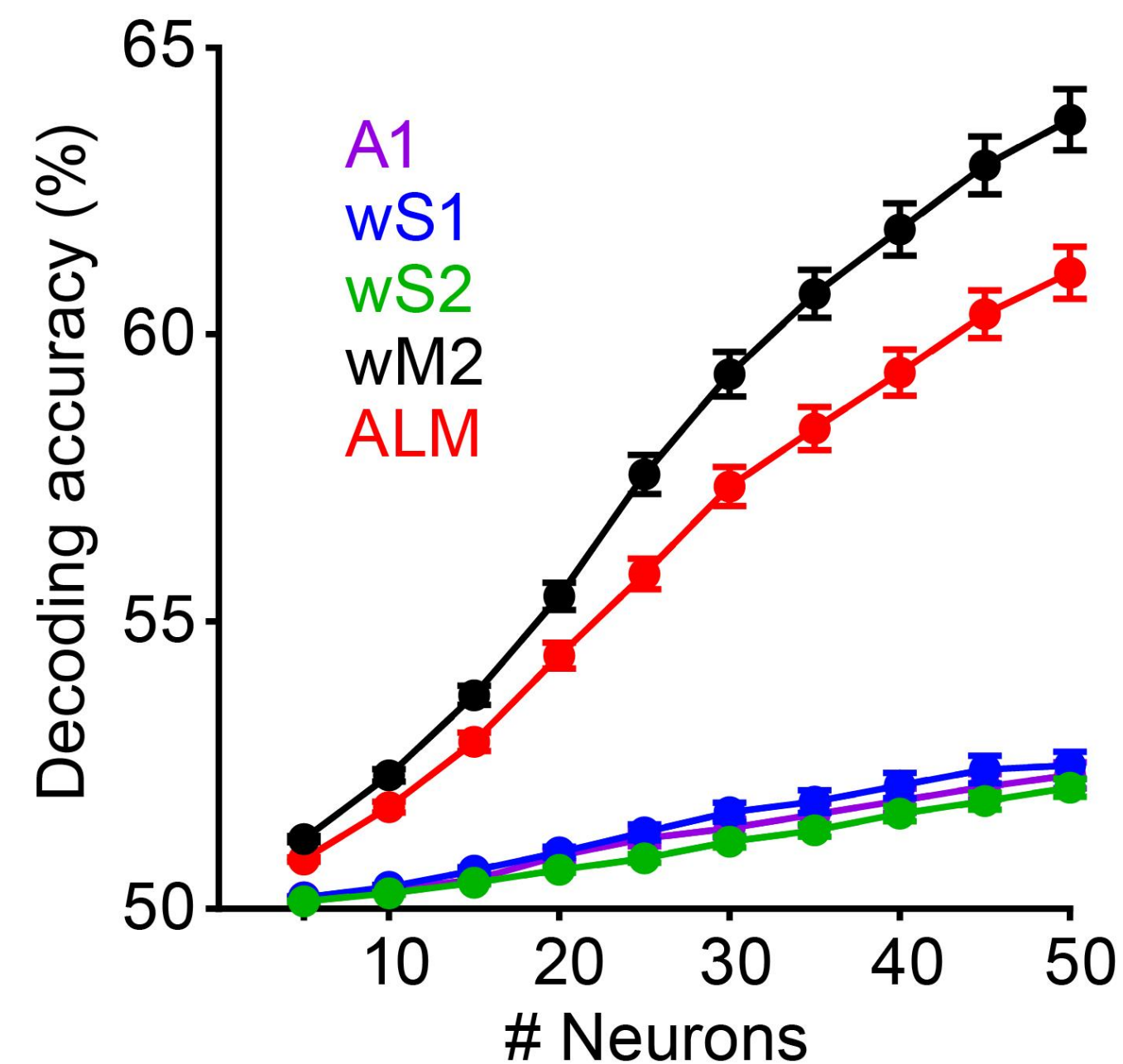


Context decoding – Go tone *vs* Nogo tone

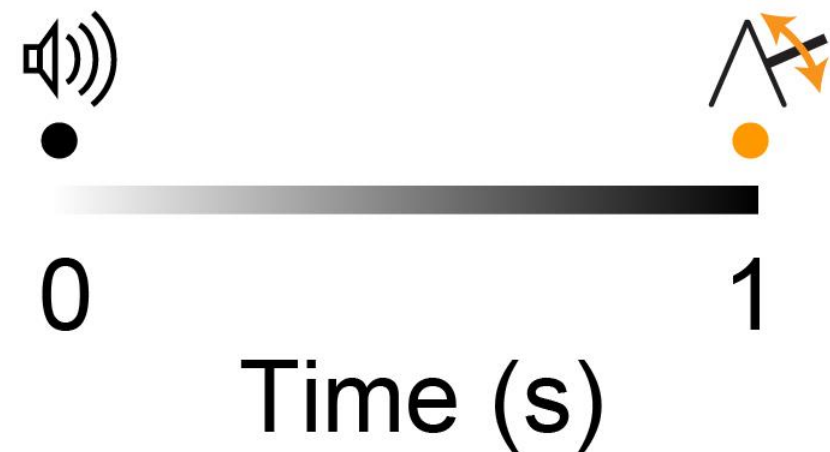
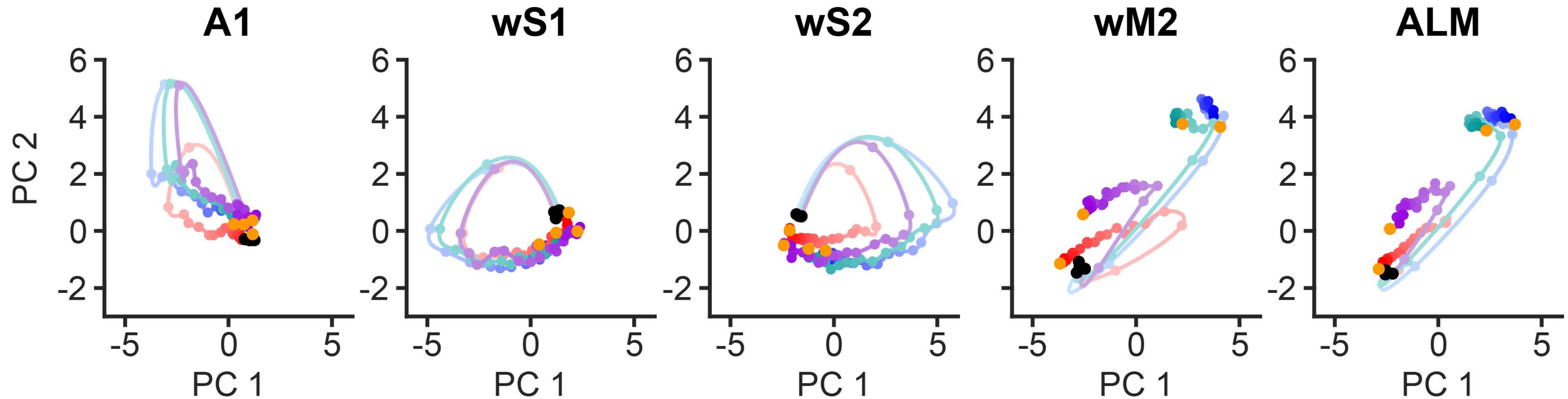
Decoding accuracy for 50 neurons per area



Decoding vs number of neurons

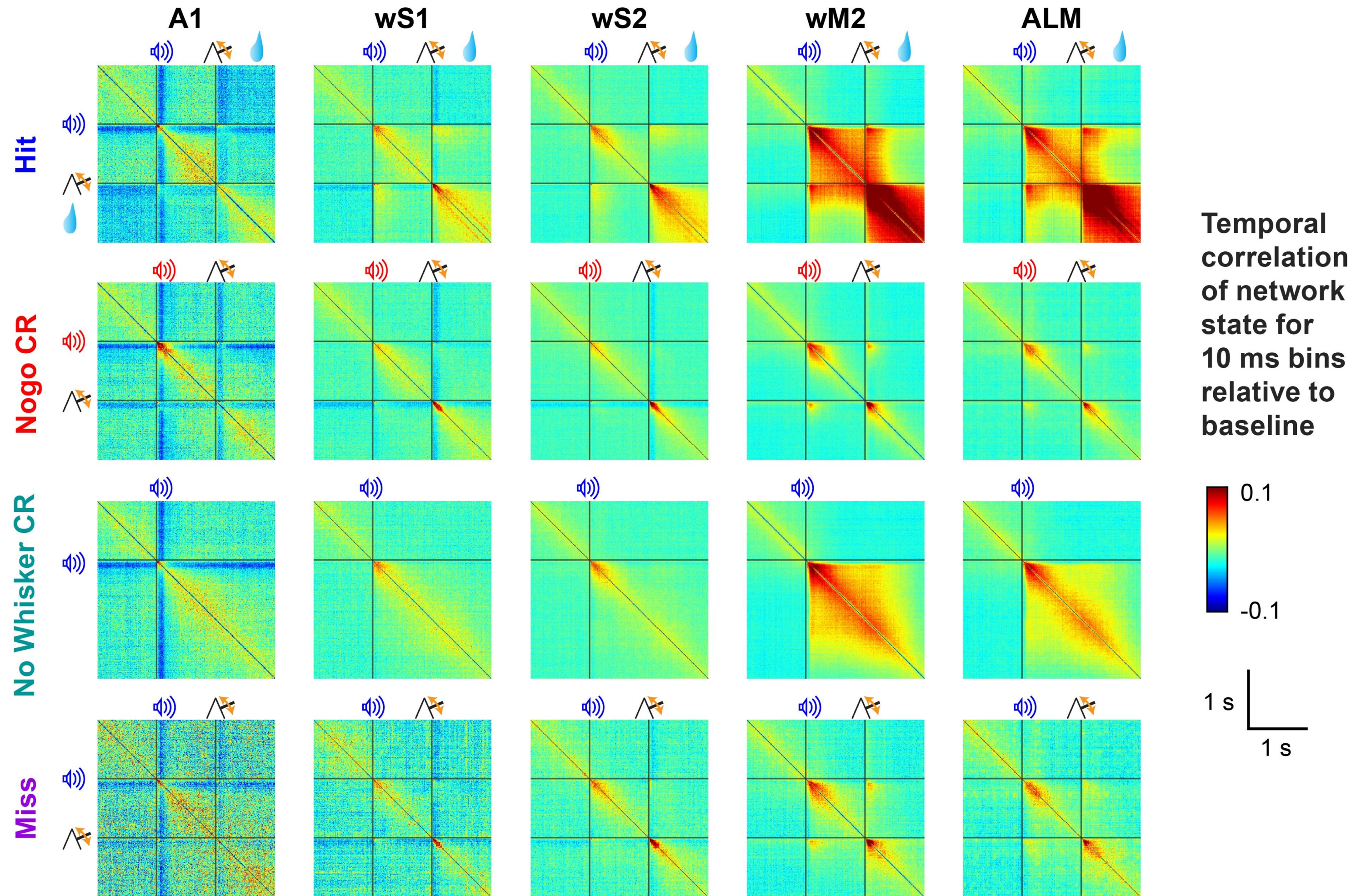


Principal component analysis

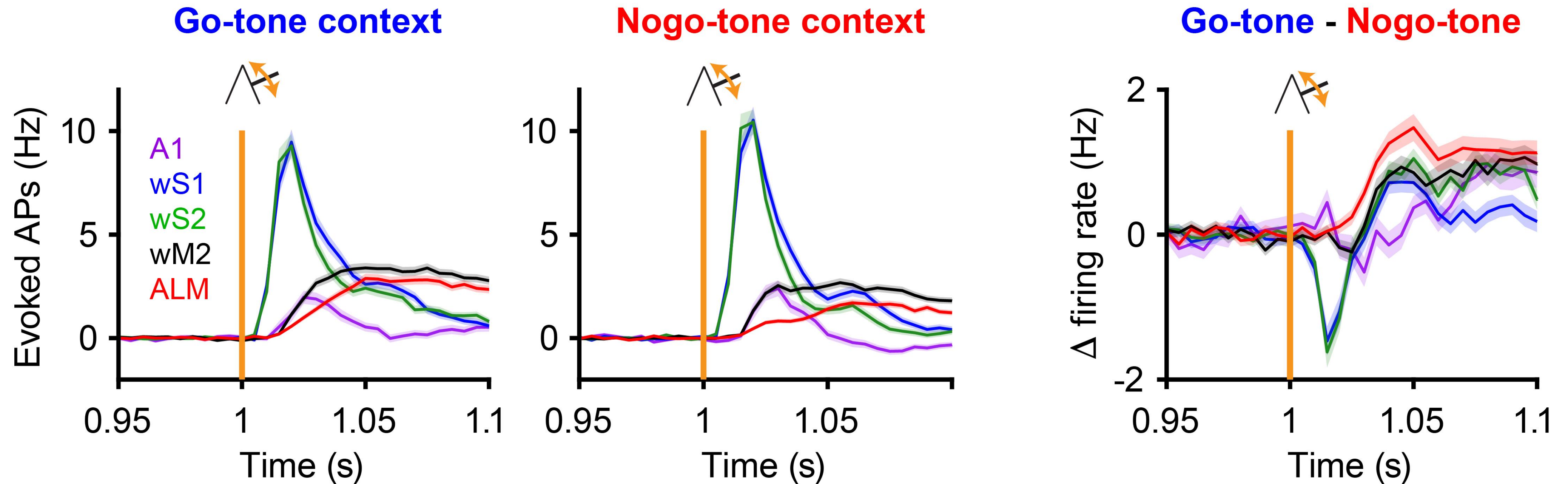


- Hit - Go-tone + Whisker + Lick
- Nogo - Nogo Tone + Whisker + No Lick
- No W - Go Tone + No Whisker + No Lick
- Miss - Go Tone + Whisker + No Lick

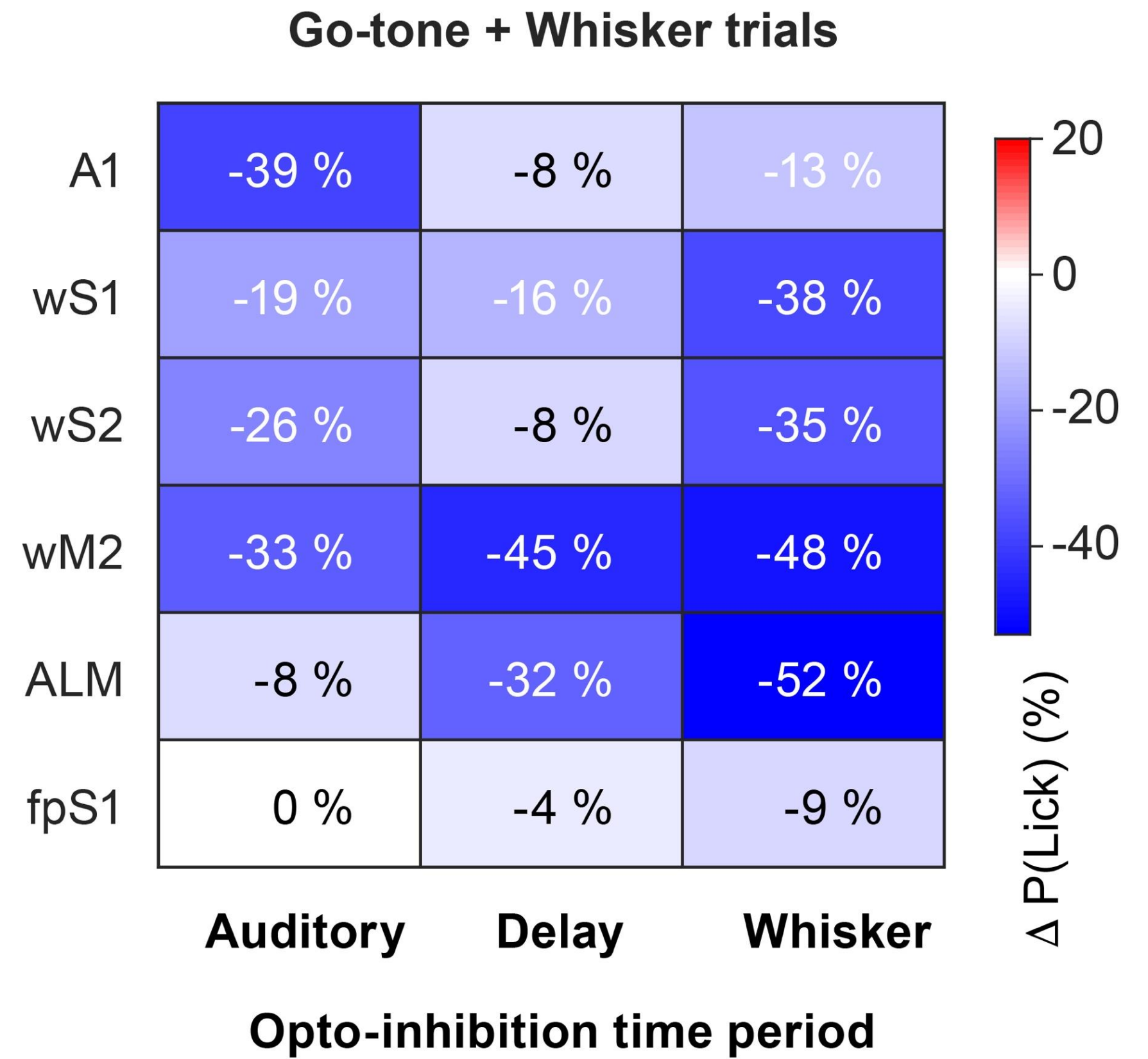
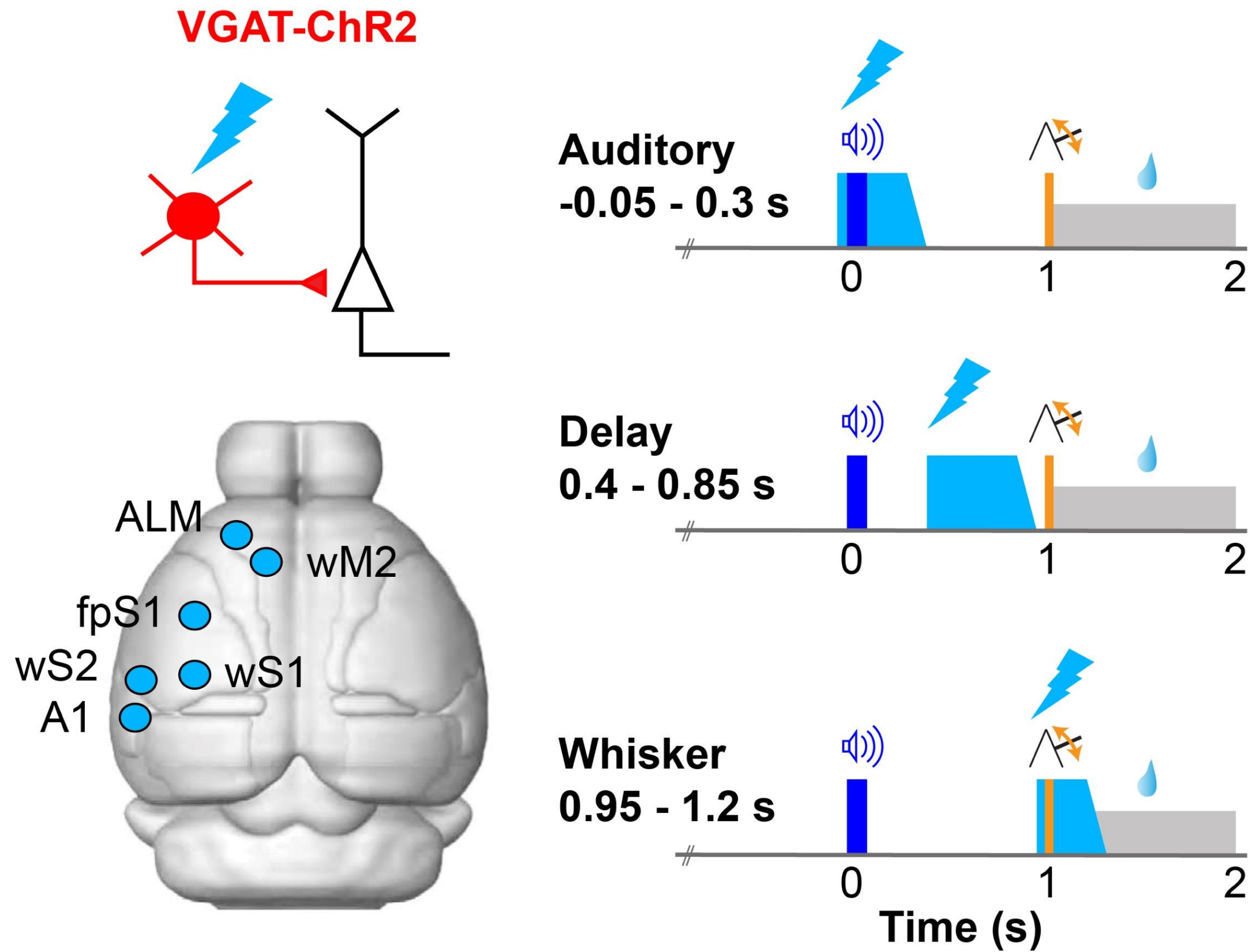
Temporal correlation of population activity



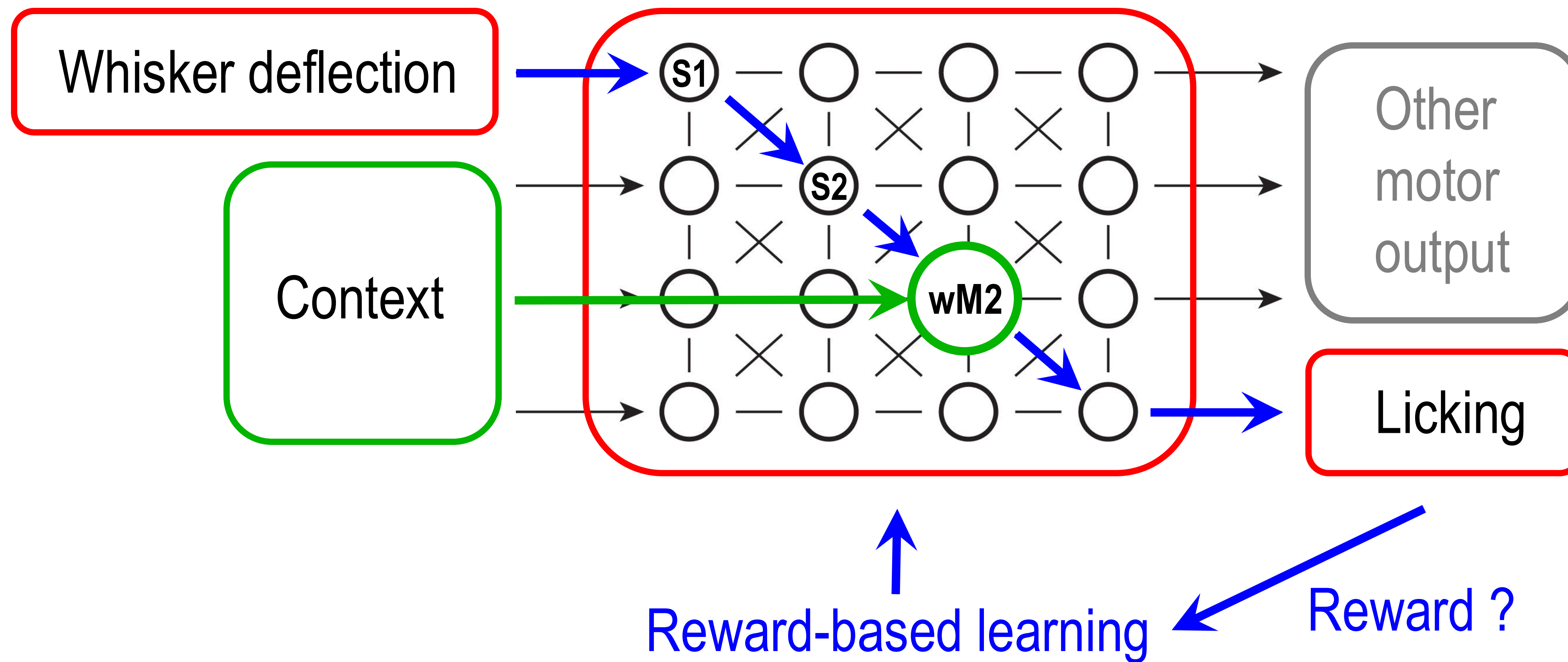
Whisker deflection-evoked response



Spatiotemporal optogenetic inactivation



Context-dependent gating by wM2 ?



NWB files

▼ EPFL_CDN	Today at 08:58	--
▼ data_electrophysiology	Today at 08:58	11.25 GB
sub-PG019_ses-20200904T224847-processed-behavior.nwb	25 June 2025 at 00:19	266 MB
sub-PG019_ses-20200906T175456-processed-behavior.nwb	25 June 2025 at 00:19	267.7 MB
sub-PG027_ses-20210403T115220-processed-behavior.nwb	25 June 2025 at 00:19	383.6 MB
sub-PG027_ses-20210404T111739-processed-behavior.nwb	25 June 2025 at 00:19	316.2 MB
sub-PG028_ses-20210318T163504-processed-behavior.nwb	25 June 2025 at 00:19	437.7 MB
sub-PG028_ses-20210319T135422-processed-behavior.nwb	25 June 2025 at 00:20	439.1 MB
sub-PG030_ses-20210405T163808-processed-behavior.nwb	25 June 2025 at 00:20	289.7 MB
sub-PG030_ses-20210406T164509-processed-behavior.nwb	25 June 2025 at 00:20	208.2 MB
sub-PG030_ses-20210407T211250-processed-behavior.nwb	25 June 2025 at 00:20	209.6 MB
sub-PG030_ses-20210408T223046-processed-behavior.nwb	25 June 2025 at 00:20	116.2 MB
sub-PG030_ses-20210410T012800-processed-behavior.nwb	25 June 2025 at 00:20	118.7 MB
sub-PG032_ses-20210711T175822-processed-behavior.nwb	25 June 2025 at 00:20	247.9 MB
sub-PG032_ses-20210712T185829-processed-behavior.nwb	25 June 2025 at 00:20	267.8 MB
sub-PG032_ses-20210715T223447-processed-behavior.nwb	25 June 2025 at 00:20	293.2 MB
sub-PG038_ses-20210316T152753-processed-behavior.nwb	25 June 2025 at 00:20	234.2 MB
sub-PG038_ses-20210317T180556-processed-behavior.nwb	25 June 2025 at 00:21	177.2 MB
sub-PG061_ses-20220211T222307-processed-behavior.nwb	25 June 2025 at 00:21	425.2 MB

Python code – Tâm Nguyen

Using a specified time binning as well as initial and final window, we will further process the arrays and dataframes into binary signals and trial-sliced tensors:

1. spikes_raster : a Time x Neurons tensor containing the unit activation (binary signal)
2. trial_spikes : a Trial x Neuron x Times tensor containing the unit activation (binary signal)
3. trial_behavior : a Trial x Behavior x Times tensor where the indexing of the behavior is specified in behavior_keys
4. behavior_keys : indexing map for the behavior in trial_behavior

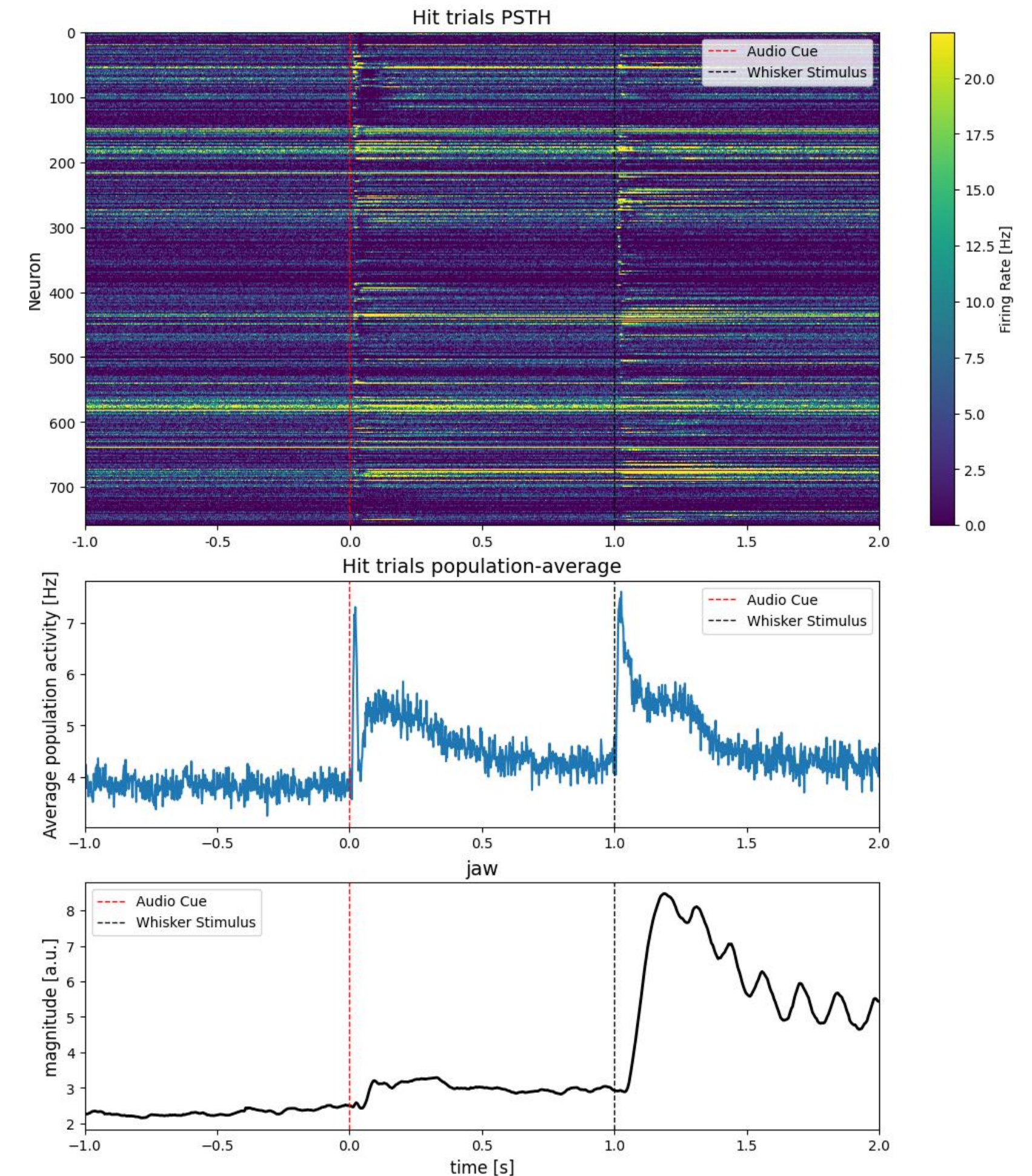
```
dt = 0.002 # 2 ms bins
dt_behavior = int(np.round(1000*(behavior_df.time[1] - behavior_df.time[0])))/1000 # rounded to nearest ms
initial_window = 1.
final_window = 0.
spikes_raster = get_raster(spike_times, spike_clusters, dt)
trial_spikes = get_trial_spikes(spikes_raster, trial_df, dt, initial_window, final_window)
trial_behavior, behavior_keys = get_trial_behavior(behavior_df, trial_df, dt_behavior, initial_window, final_window)
```

```
def compute_PSTH(trial_spikes, trial_mask=None):
    return np.mean(trial_spikes, axis=0) if trial_mask is None else np.mean(trial_spikes[trial_mask], axis=0)
```

```
time = np.linspace(-initial_window, 2+final_window, trial_spikes.shape[-1])
time_behavior = np.linspace(-initial_window, 2+final_window, trial_behavior.shape[-1])
areas = cluster_df.area.unique()
```

```
trial_mask_hit = np.logical_and(trial_df['trial_type'] == 1, trial_df['lick'] == 1)
psth_hit = compute_PSTH(trial_spikes, trial_mask_hit)
```

```
plt.figure(figsize=(10, 6))
for area in areas:
    plt.plot(time, psth_hit[cluster_df.area == area].mean(axis=0), label = area)
max_value = max([psth_hit[cluster_df.area == area].mean(axis=0).max() for area in areas])
plt.vlines([0, 1], ymin = 0, ymax = max_value + 0.001, colors='k', linestyle='dashed')
plt.xlabel("time [s]")
plt.ylabel("firing rate [Hz]")
plt.title("Grand PSTH for Hit trials per area")
plt.xlim([-initial_window, 2+final_window])
plt.ylim([0, max_value + 0.001])
plt.legend()
plt.show()
```



Thank you!



EPFL

This presentation

Parviz Ghaderi

Sylvain Crochet

Current lab members

Pol Bech

Axel Bisi

Sylvain Crochet

Robin Dard

Parviz Ghaderi

Myriam Hamon

Jules Lebert

Meriam Malekzadeh

Tâm Nguyen

Mauro Pulin

Anthony Renard

Shankar Sachidhanandam

Lana Smith

Morgane Storey

Aleksandra Teska