# S021 analysis

MeganePro Dataset 1, subject S021 (able-bodied, 2% invalid gaze samples). Sections: trial segmentation, gaze timing, table layouts, camera-frame classifier, object-relative gaze, onset comparison, detector coverage. Inputs are in `features/` and `layout_check/`.

In [1]:

```
import os, json, re, numpy as np, pandas as pd, matplotlib.pyplot as plt, cv2
os.chdir(os.path.expanduser('~/mnt/thesis-data/meganepro'))
plt.rcParams.update({'figure.dpi': 110, 'axes.spines.top': False, 'axes.spines.right': False})
ev = pd.read_csv('features/S021_events.csv'); st = ev[ev.dynamic == 0].reset_index(drop=True)
LAYOUT = {"medium wrap":"A","lateral":"A","parallel extension":"B","tripod":"B","power sphere":"C","precision
disk":"C","prismatic pinch":"C","index finger extension":"D","adducted thumb":"D","prismatic four finger":"D"}
st['layout'] = st.grasp_name.map(LAYOUT)
objrel = pd.read_csv('features/S021_objrel.csv')
det = {json.loads(l)['frame']: json.loads(l)['dets'] for l in open('layout_check/detections.jsonl')}
def show(paths, titles, ncol=3, h=3.2, boxes=None, gaze=None):
    n = len(paths); nrow = int(np.ceil(n/ncol)); fig, axs = plt.subplots(nrow, ncol, figsize=(ncol*4.6, nrow*h));
axs = np.array(axs).ravel()
    for i, (p, t) in enumerate(zip(paths, titles)):
        im = cv2.cvtColor(cv2.imread(p), cv2.COLOR_BGR2RGB); axs[i].imshow(im); axs[i].set_title(t, fontsize=9);
axs[i].axis('off')
        if boxes is not None and boxes[i] is not None:
            x1,y1,x2,y2 = boxes[i]; axs[i].add_patch(plt.Rectangle((x1,y1),x2-x1,y2-
y1,fill=False,color='lime',lw=2))
        if gaze is not None and gaze[i] is not None and not np.isnan(gaze[i][0]):
            axs[i].plot(gaze[i][0]*im.shape[1], gaze[i][1]*im.shape[0], 'o', ms=9, mfc='none', mec='red', mew=2)
    for a in axs[n:]: a.axis('off')
    plt.tight_layout(); plt.show()
print(f'{len(st)} static trials, {st.object_name.nunique()} objects, {st.grasp_name.nunique()} grips')
```

```
240 static trials, 18 objects, 10 grips
```

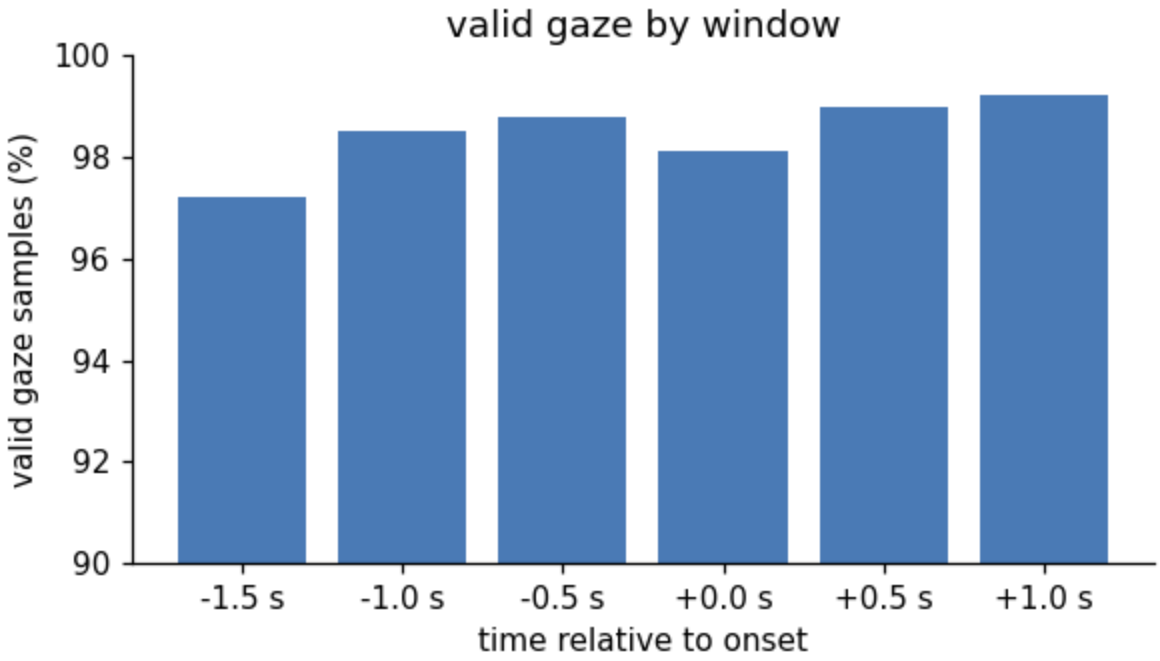
# 1. Trial segmentation and gaze validity

Trials cut from the stimulus labels: onset = end of voice command, end = rest command. Practice trials dropped. Protocol specifies 8 static trials per object-grip pair. Gaze validity measured in a 100 ms window at each time point relative to onset.

In [2]:

```
t1 = st.groupby('object_name').agg(grips=('grasp_name', lambda s: ', '.join(sorted(set(s)))), n_grips=('grasp',
'nunique'), trials=('grasp', 'size')).sort_values(['n_grips','object_name'], ascending=[False,True])
display(t1)
win = pd.read_csv('features/S021_summary_windows.csv')
fig, ax = plt.subplots(figsize=(6,3))
ax.bar(win.window, win.valid_frac*100, color='#4a7ab5'); ax.set_ylim(90,100); ax.set_ylabel('valid gaze samples
(%)'); ax.set_xlabel('time relative to onset')
ax.set_title('valid gaze by window'); plt.show()
```

| object_name | grips | n_grips | trials |
|---|---|---|---|
| **key** | lateral, power sphere, prismatic pinch | 3 | 24 |
| **ball** | power sphere, precision disk | 2 | 16 |
| **bottle** | medium wrap, tripod | 2 | 16 |
| **bulb** | power sphere, precision disk | 2 | 16 |
| **can** | medium wrap, prismatic pinch | 2 | 16 |
| **drawer** | parallel extension, tripod | 2 | 16 |
| **fork** | index finger extension, prismatic four finger | 2 | 16 |
| **knife** | index finger extension, prismatic four finger | 2 | 16 |
| **mug** | lateral, tripod | 2 | 16 |
| **remote** | adducted thumb, index finger extension | 2 | 16 |
| **wrench** | adducted thumb, prismatic four finger | 2 | 16 |
| **book** | parallel extension | 1 | 8 |
| **clothespin** | prismatic pinch | 1 | 8 |
| **door handle** | medium wrap | 1 | 8 |
| **jar** | precision disk | 1 | 8 |
| **pencil case** | lateral | 1 | 8 |
| **plate** | parallel extension | 1 | 8 |
| **screwdriver** | adducted thumb | 1 | 8 |



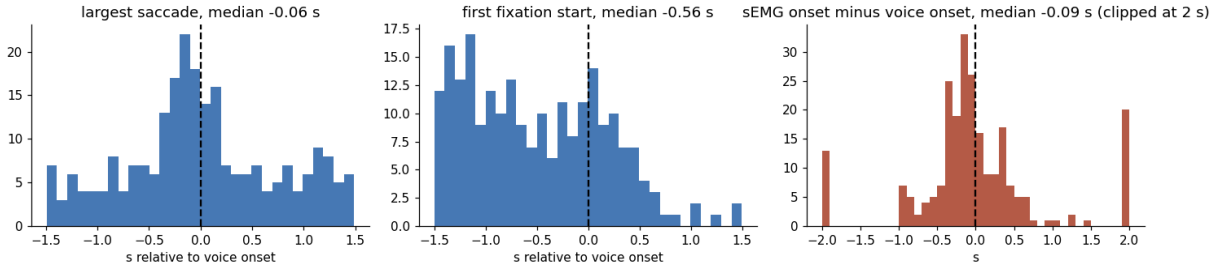

240 trials, 8 per pair. 11 of 18 objects have more than one grip; the 7 single-grip objects are excluded below.

## 2. Gaze timing

Largest saccade in the 3 s around onset; first fixation in the 1.5 s before onset; sEMG-realigned onset minus voice onset.

In [3]:

```
fig, axs = plt.subplots(1, 3, figsize=(13,3.2))
axs[0].hist(st.sacc_peak_t.dropna(), bins=30, color='#4a7ab5'); axs[0].axvline(0, color='k', ls='--');
axs[0].set_title(f'largest saccade, median {st.sacc_peak_t.median():.2f} s'); axs[0].set_xlabel('s relative to
voice onset')
axs[1].hist(st.first_fix_t.dropna(), bins=30, color='#4a7ab5'); axs[1].axvline(0, color='k', ls='--');
axs[1].set_title(f'first fixation start, median {st.first_fix_t.median():.2f} s'); axs[1].set_xlabel('s relative
to voice onset')
re_ = pd.read_csv('features/S021_onsets_re.csv'); re_ = re_[re_.dynamic==0]
sh = (re_.t_on_re - re_.t_on).clip(-2, 2)
axs[2].hist(sh, bins=40, color='#b55a4a'); axs[2].axvline(0, color='k', ls='--'); axs[2].set_title(f'sEMG onset
minus voice onset, median {sh.median():.2f} s (clipped at 2 s)'); axs[2].set_xlabel('s')
plt.tight_layout(); plt.show()
print(f'{(st.sacc_peak_t<0).mean():.0%} of trials: largest saccade before onset. Head angular speed in the second
before onset: median {st.head_gyr_pre.median():.1f} deg/s.')
```




```
54% of trials: largest saccade before onset. Head angular speed in the second before onset: median 10.0 deg/s.
```

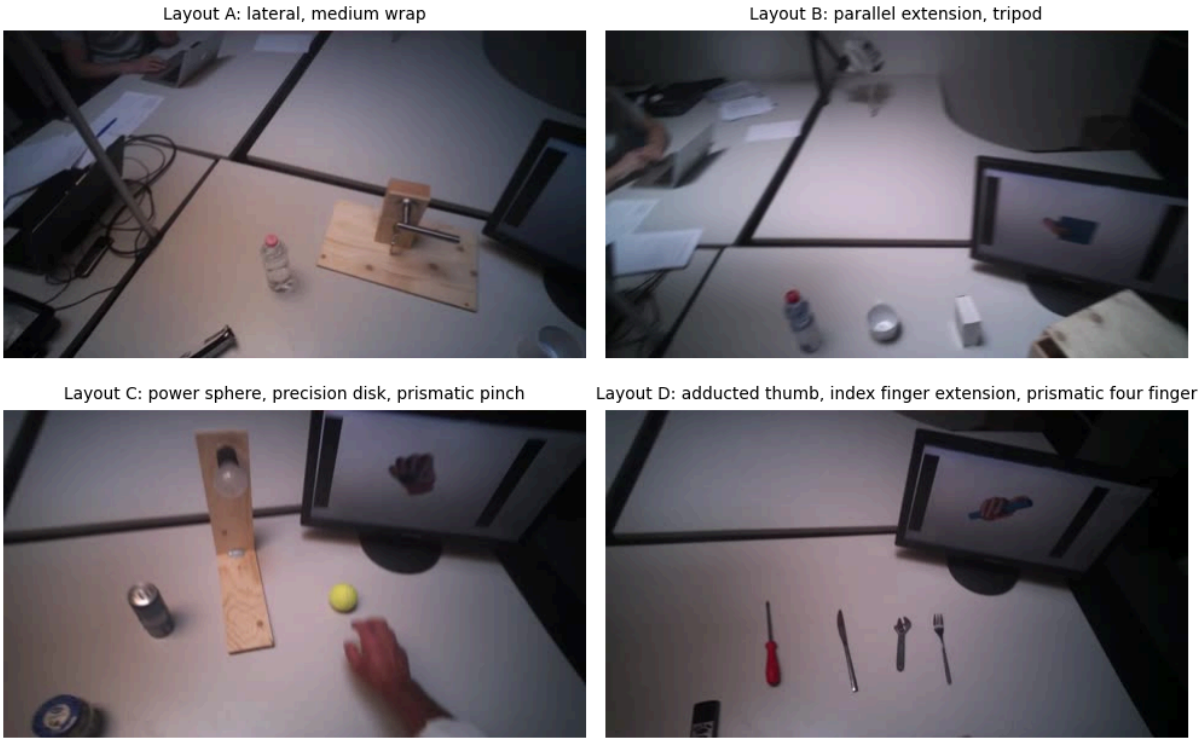
Gaze reaches the target before the voice command ends. sEMG onset: IQR -0.33 to +0.29 s relative to voice onset, outliers to -10 and +5 s.

# 3. Table layouts

Four table arrangements across the ten grip blocks (from the video). One frame per layout at onset + 0.3 s.

In [4]:

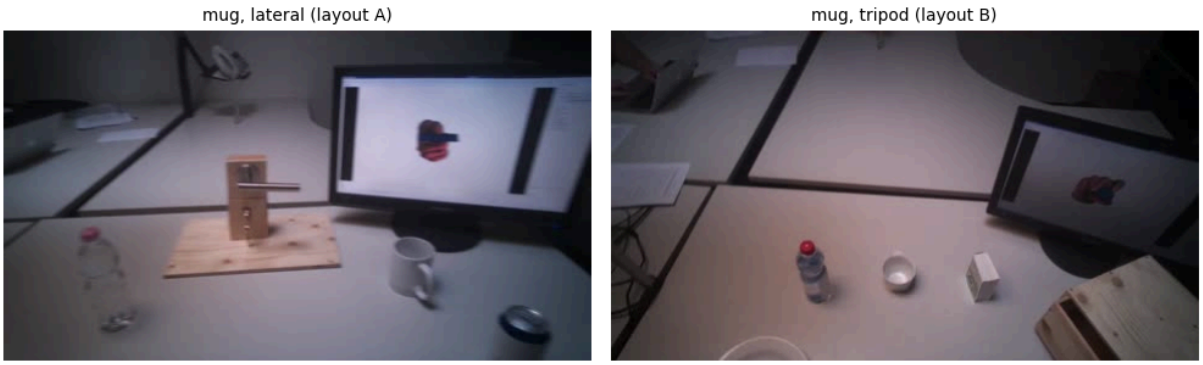
```
idx = pd.read_csv('layout_check/frame_index.csv'); files = sorted(os.listdir('layout_check/frames'))
picks = []
for lay, grp in st.groupby('layout'):
    k = grp.index[len(grp)//4]; f = [x for x in files if x.startswith(f'{k:03d}_')][0]
    picks.append(('layout_check/frames/'+f, f'Layout {lay}: ' + ', '.join(sorted(set(grp.grasp_name)))))
show([p for p,_ in picks], [t for _,t in picks], ncol=2, h=3.0)
```




Objects stay in place within a layout; the table is rearranged between layouts. Grips were run as blocks. For an object whose two grips fall in different layouts, camera-frame gaze position is confounded with object position. Mug: layout A (lateral, handle visible), layout B (tripod, handle not visible).

In [5]:

```
mug = st[st.object_name=='mug']
picks = [(f'layout_check/frames/'+[x for x in files if x.startswith(f'{k:03d}_')][0], f'mug, {r.grasp_name}
(layout {r.layout})') for k, r in [(mug.index[0], mug.iloc[0]), (mug.index[-1], mug.iloc[-1])]]
show([p for p,_ in picks], [t for _,t in picks], ncol=2, h=3.0)
```



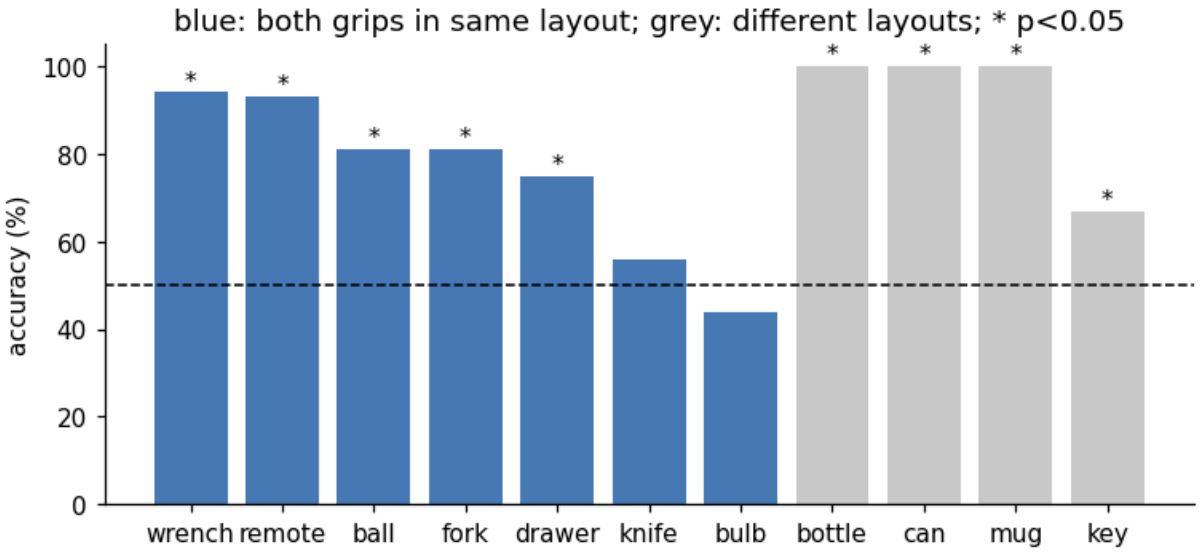

# 4. Camera-frame gaze vs grip

Per object: logistic regression on gaze (x, y) at -0.5, 0, +0.5 s plus posture; leave-one-repetition-out; permutation test, 200 shuffles of the grip label within object. Flag = whether both grips were recorded in the same layout.

In [6]:

```
t4 = pd.read_csv('features/S021_summary_classifier_layout.csv')
display(t4[['object','grips','layouts','same_layout','majority_baseline','gaze_acc','perm_p']])
fig, ax = plt.subplots(figsize=(8,3.4))
cols = t4.same_layout.map({'yes':'#4a7ab5','no':'#c9c9c9'})
ax.bar(t4.object, t4.gaze_acc*100, color=cols); ax.axhline(50, color='k', ls='--', lw=1)
for i,(a,p) in enumerate(zip(t4.gaze_acc, t4.perm_p)): ax.text(i, a*100+1, '*' if p<0.05 else '', ha='center')
ax.set_ylabel('accuracy (%)'); ax.set_title('blue: both grips in same layout; grey: different layouts; * p<0.05');
plt.show()
```

| | object | grips | layouts | same_layout | majority_baseline | gaze_acc | perm_p |
|---|---|---|---|---|---|---|---|
| 0 | wrench | adducted thumb, prismatic four finger | D | yes | 0.50 | 0.94 | 0.005 |
| 1 | remote | index finger extension, adducted thumb | D | yes | 0.53 | 0.93 | 0.010 |
| 2 | ball | power sphere, precision disk | C | yes | 0.50 | 0.81 | 0.020 |
| 3 | fork | index finger extension, prismatic four finger | D | yes | 0.50 | 0.81 | 0.015 |
| 4 | drawer | parallel extension, tripod | B | yes | 0.50 | 0.75 | 0.045 |
| 5 | knife | index finger extension, prismatic four finger | D | yes | 0.50 | 0.56 | 0.318 |
| 6 | bulb | power sphere, precision disk | C | yes | 0.50 | 0.44 | 0.617 |
| 7 | bottle | medium wrap, tripod | A/B | no | 0.50 | 1.00 | 0.005 |
| 8 | can | medium wrap, prismatic pinch | A/C | no | 0.50 | 1.00 | 0.005 |
| 9 | mug | lateral, tripod | A/B | no | 0.50 | 1.00 | 0.005 |
| 10 | key | lateral, power sphere, prismatic pinch | A/C | no | 0.33 | 0.67 | 0.005 |




Cross-layout objects: bottle, can, mug (100%), key (67%). Same-layout: wrench 94, remote 93, ball 81, fork 81, drawer 75 above chance; knife 56, bulb 44 not. Ball and bulb share the same grip pair and block pair.

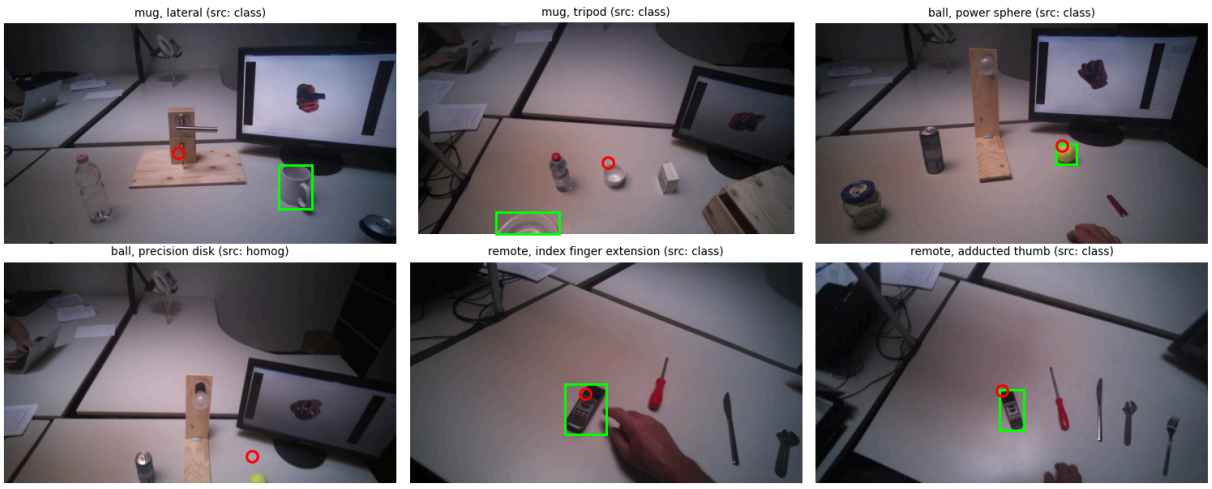
## 5. Object-relative gaze

Gaze expressed as (u, v) inside the target object's bounding box. Boxes from YOLO-World (zero-shot, 18 class prompts) on the frame at each window. If the target class is not detected with confidence >= 0.25 in any window: box = the detection containing

gaze at 0 or +0.5 s. Remaining windows: box carried from the nearest boxed window by SIFT homography (>= 15 inliers). Green = box used, red = gaze.

In [7]:

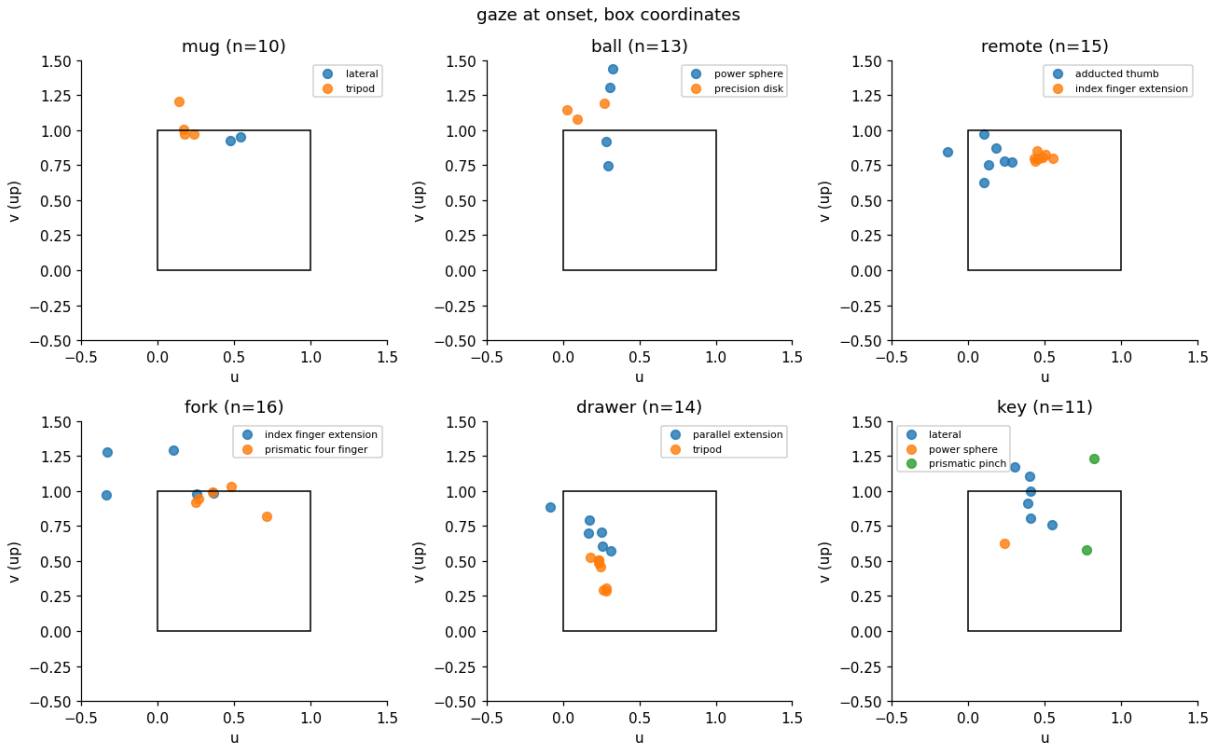
```
def box_for(row, w='+0.0'):
    k, on = int(row.trial), row.onset; f = f'{k:03d}_{on}{w}.jpg'
    x, y = row[f'gx{w}']*960, row[f'gy{w}']*540
    cands = det.get(f, [])
    if np.isnan(row[f'u{w}']): return f, None
    # reconstruct the box from u,v is not possible; pick the detection that reproduces u,v
    for d in cands:
        x1,y1,x2,y2 = d['box']
        if abs((x-x1)/max(x2-x1,1)-row[f'u{w}'])<1e-3 and abs((y-y1)/max(y2-y1,1)-row[f'v{w}'])<1e-3: return f,
d['box']
    return f, None
ex = objrel[(objrel.onset=='v')].dropna(subset=['u+0.0'])
rows = [ex[ex.object==o].iloc[i] for o,i in [('mug',0),('mug',-1),('ball',0),('ball',-1),('remote',0),
('remote',-1)]]
paths, titles, boxes, gaze = [], [], [], []
for r in rows:
    f, b = box_for(r); paths.append('layout_check/winframes/'+f); boxes.append(b); gaze.append((r['gx+0.0'],
r['gy+0.0']))
    titles.append(f"{r.object}, {r.grasp} (src: {re.sub(r'homog.*','homog', r['src+0.0'])})")
show(paths, titles, ncol=3, h=2.8, boxes=boxes, gaze=gaze)
```




Gaze at onset in box coordinates, per object, by grip. (u, v) in [0, 1] across the box.

In [8]:

```
objs = ['mug','ball','remote','fork','drawer','key']
fig, axs = plt.subplots(2, 3, figsize=(12, 7.5)); axs = axs.ravel()
for ax, o in zip(axs, objs):
    d = objrel[(objrel.onset=='v')&(objrel.object==o)].dropna(subset=['u+0.0'])
    for g, dd in d.groupby('grasp'): ax.scatter(dd['u+0.0'], 1-dd['v+0.0'], label=g, s=40, alpha=0.8)
    ax.add_patch(plt.Rectangle((0,0),1,1,fill=False,color='k')); ax.set_xlim(-0.5,1.5); ax.set_ylim(-0.5,1.5);
ax.set_title(f'{o} (n={len(d)})'); ax.legend(fontsize=7); ax.set_xlabel('u'); ax.set_ylabel('v (up)')
plt.suptitle('gaze at onset, box coordinates'); plt.tight_layout(); plt.show()
```

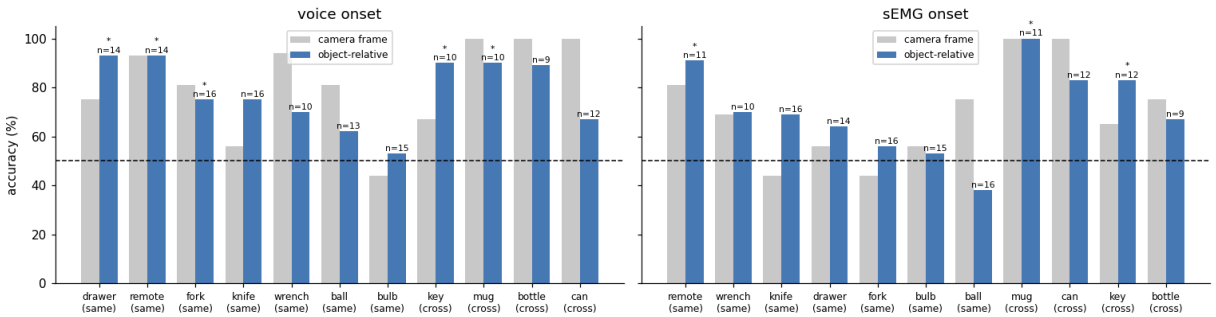



# 6. Camera-frame vs object-relative; voice vs sEMG onset

Same classifier on (u, v) features. Left: windows relative to voice onset. Right: relative to sEMG onset. n_rel = trials with a box.

In [9]:

```
fig, axs = plt.subplots(1, 2, figsize=(14, 3.8), sharey=True)
for ax, on, name in zip(axs, ['v','e'], ['voice onset', 'sEMG onset']):
    t = pd.read_csv(f'features/S021_summary_classifier_objrel_{on}.csv')
    x = np.arange(len(t)); w = 0.38
    ax.bar(x-w/2, t.acc_cam*100, w, label='camera frame', color='#c9c9c9'); ax.bar(x+w/2, t.acc_rel*100, w,
label='object-relative', color='#4a7ab5')
    for i,(a,p,n) in enumerate(zip(t.acc_rel, t.perm_p_rel, t.n_rel)): ax.text(i+w/2, a*100+1, ('*' if p<0.05 else
'')+f'\nn={n}', ha='center', fontsize=7)
    ax.set_xticks(x); ax.set_xticklabels([f"{o}\n({'same' if s=='yes' else 'cross'})" for o,s in zip(t.object,
t.same_layout)], fontsize=8)
    ax.axhline(50, color='k', ls='--', lw=1); ax.set_title(name); ax.legend(fontsize=8)
axs[0].set_ylabel('accuracy (%)'); plt.tight_layout(); plt.show()
display(pd.read_csv('features/S021_summary_classifier_objrel_v.csv'))
```



| | object | layouts | same_layout | n_cam | n_rel | baseline_cam | acc_cam | perm_p_cam | acc_rel | perm_p_rel |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | drawer | B | yes | 16 | 14 | 0.50 | 0.75 | 0.025 | 0.93 | 0.005 |
| **1** | remote | D | yes | 15 | 14 | 0.53 | 0.93 | 0.005 | 0.93 | 0.005 |
| **2** | fork | D | yes | 16 | 16 | 0.50 | 0.81 | 0.035 | 0.75 | 0.025 |
| **3** | knife | D | yes | 16 | 16 | 0.50 | 0.56 | 0.318 | 0.75 | 0.050 |
| **4** | wrench | D | yes | 16 | 10 | 0.50 | 0.94 | 0.005 | 0.70 | 0.303 |
| **5** | ball | C | yes | 16 | 13 | 0.50 | 0.81 | 0.025 | 0.62 | 0.284 |
| **6** | bulb | C | yes | 16 | 15 | 0.50 | 0.44 | 0.582 | 0.53 | 0.398 |
| **7** | key | A/C | no | 24 | 10 | 0.33 | 0.67 | 0.015 | 0.90 | 0.020 |
| **8** | mug | A/B | no | 16 | 10 | 0.50 | 1.00 | 0.005 | 0.90 | 0.025 |
| **9** | bottle | A/B | no | 16 | 9 | 0.50 | 1.00 | 0.005 | 0.89 | 0.055 |
| **10** | can | A/C | no | 16 | 12 | 0.50 | 1.00 | 0.005 | 0.67 | 0.234 |

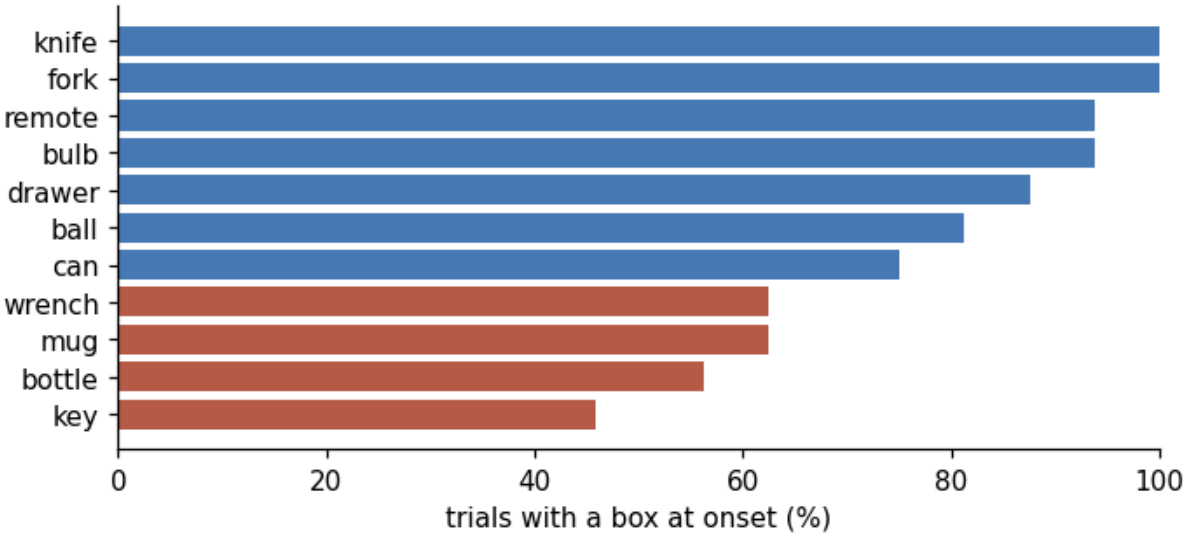
Same-layout, object-relative, voice onset: drawer 93, remote 93, fork 75, knife 75 ($p < 0.05$). Wrench 70 (n=10) and ball 62 (n=13) lose trials to missed detections. Cross-layout objects: each grip recorded in one layout only, so grip and layout are the same label within this subject; not separable without other subjects. sEMG onset: all objects except remote lower than with voice onset; ball at chance.

# 7. Detector coverage

Trials with a box at onset, per object.

In [10]:

```
cov = objrel[objrel.onset=='v'].groupby('object')['src+0.0'].apply(lambda s: (s!='none').mean()*100).sort_values()
fig, ax = plt.subplots(figsize=(7,3)); ax.barh(cov.index, cov.values, color=['#b55a4a' if v<70 else '#4a7ab5' for
v in cov.values]); ax.set_xlabel('trials with a box at onset (%)'); ax.set_xlim(0,100); plt.show()
src = objrel[objrel.onset=='v']['src+0.0'].map(lambda x: re.sub(r'homog\d+','homog',x)).value_counts(); print('box
source at onset:', src.to_dict())
```




```
box source at onset: {'class': 65, 'gaze': 44, 'none': 43, 'homog': 32}
```

# Next

1. Box coverage for bottle, mug, key, wrench: fine-tune on hand-labeled frames or improve homography references.
2. Cross-subject: train on 29 able-bodied subjects, test on the held-out one; Model A (object) vs Model B (object + gaze) at each window.
3. Amputees; HOT3D.