

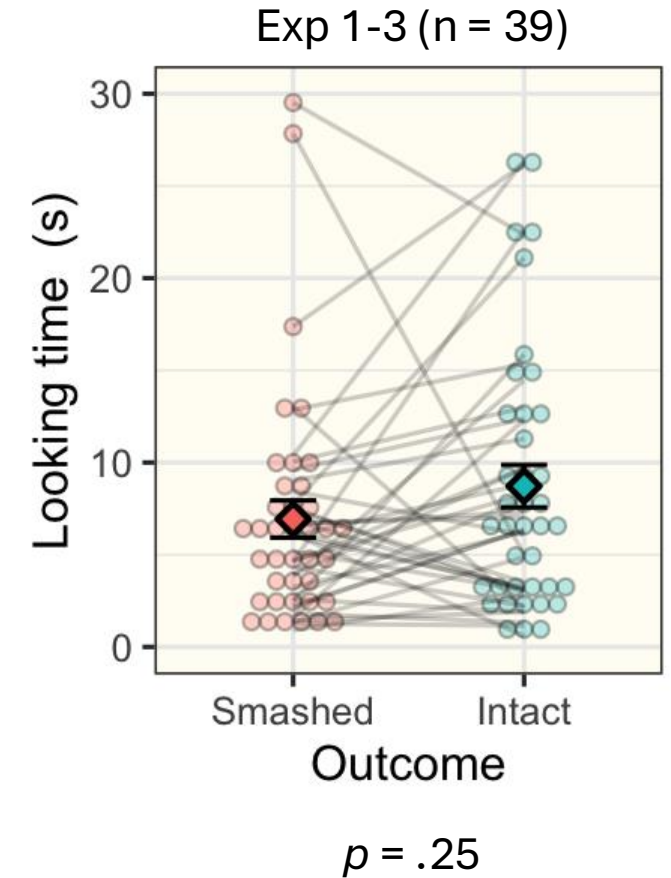
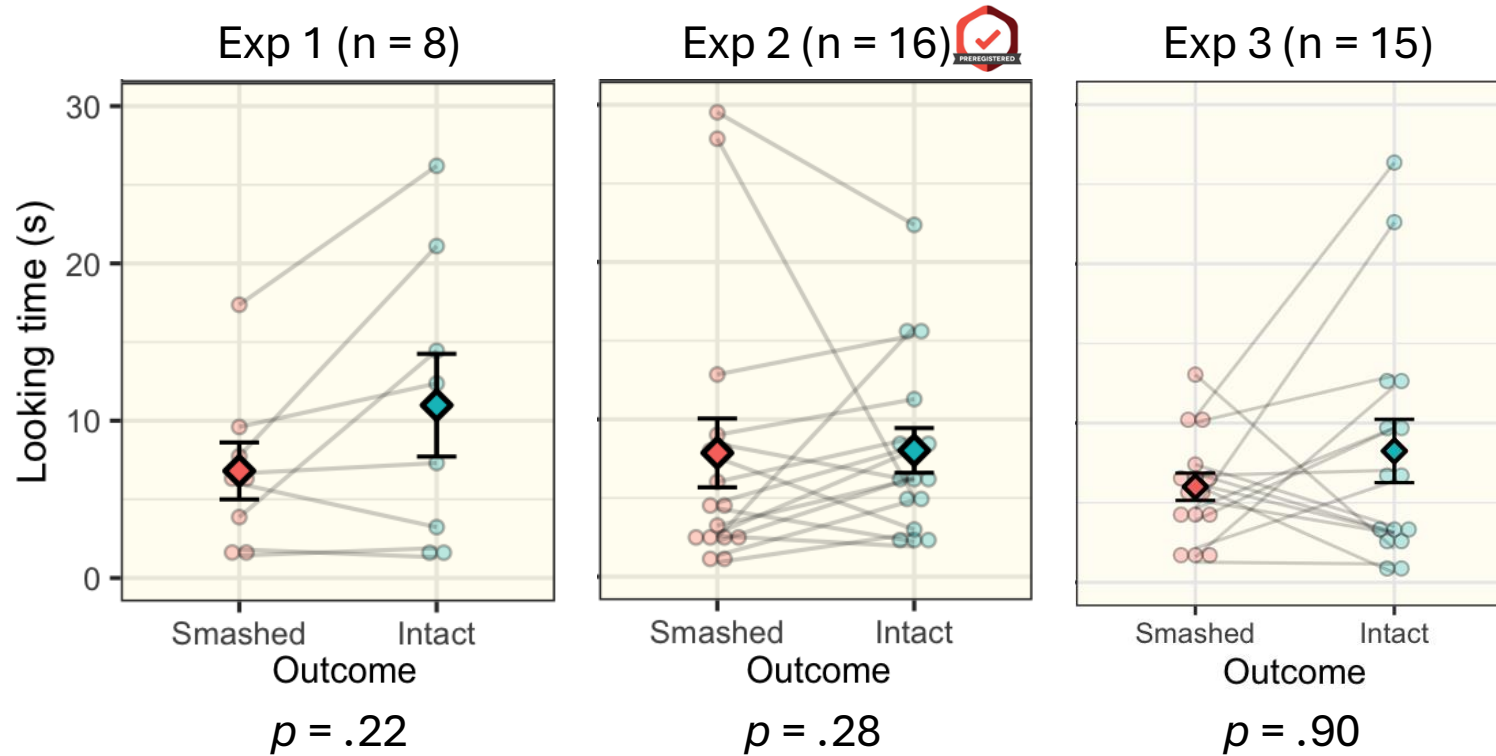
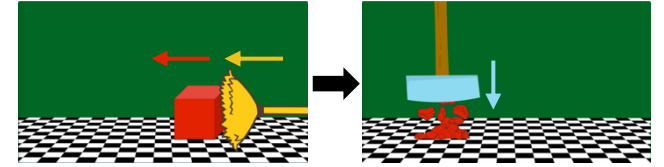
# **Supporting Information for:**

## Compositional Thought Before Compositional Language: Evidence From 9–11-Month-Olds

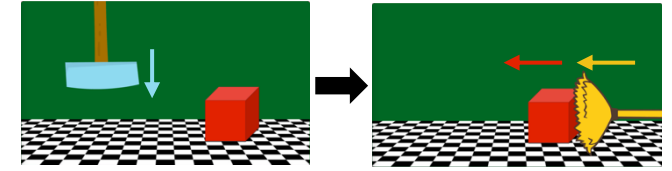
Lily Zihui Zhu, Erika Bergelson, Jesse Snedeker

Poster at BCCCD 2026

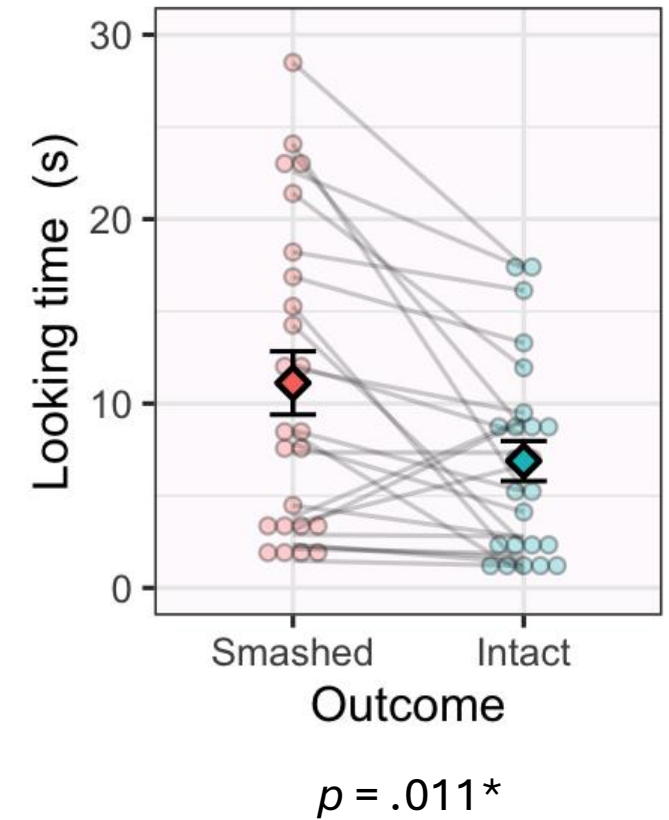
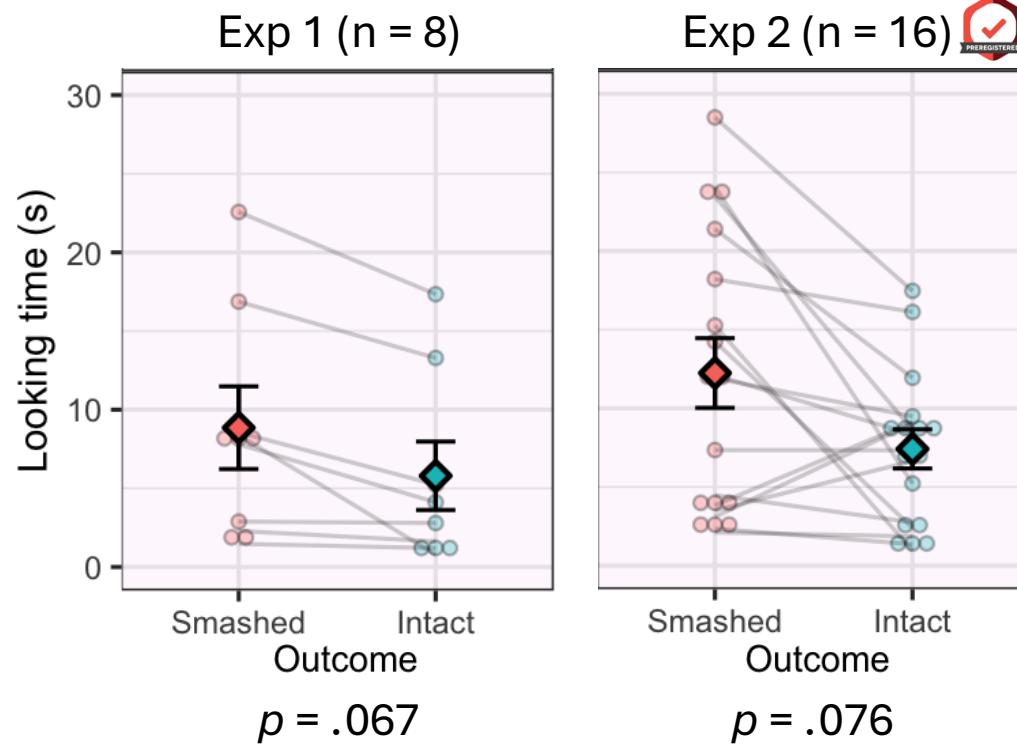
# Infants' failure in Push-First



# Infants' success in Smash-First

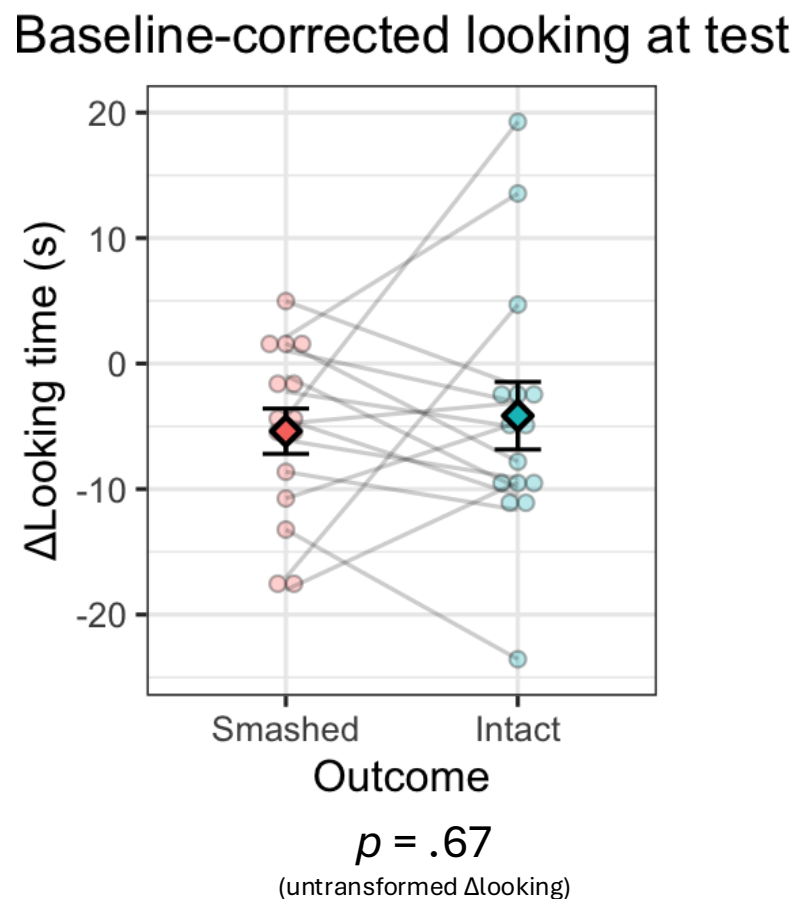
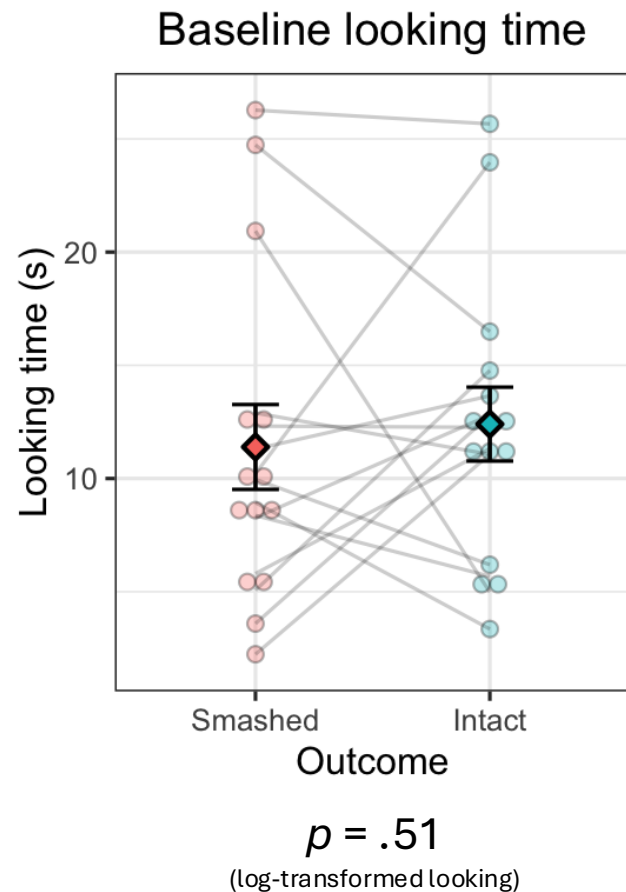


Exp 1-2 (n = 24)



# Why did 9–11-mos fail at Push-First but succeed in Smash-First?

## 1. Baseline preference: Smashed is more salient than Intact



Exp 3 (n = 15)

- Added trials to test baseline preference for each of the two outcomes at the beginning
- Subtracted baseline looking from looking at test
- **No evidence for baseline preference**

# Why did 9–11-mos fail at Push-First but succeed in Smash-First?

1. Baseline preference: Smashed is more salient than Intact

No baseline preference (Exp 3)

2. Outcome bias: Unexpected effect (smashed when shouldn't) is more salient than unexpected null effect (intact when shouldn't)

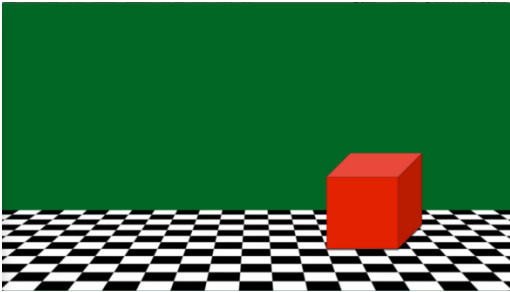
Test infants on single *smash()* functions

3. Working memory constraints: Push-First may be harder than Smash-First

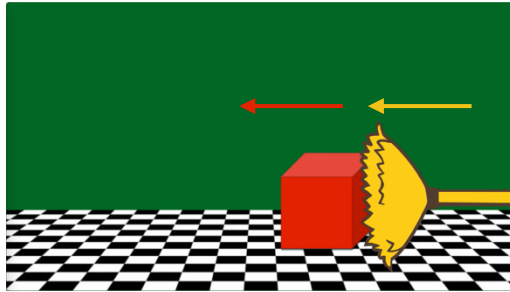
# Is Push-First harder than Smash-First?

## Push-First

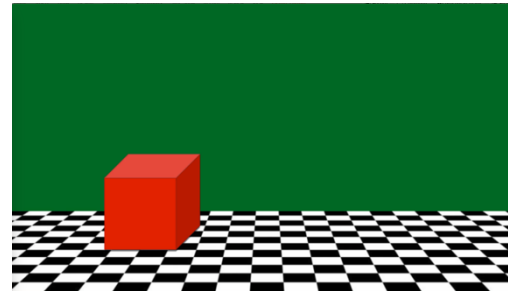
Initial state



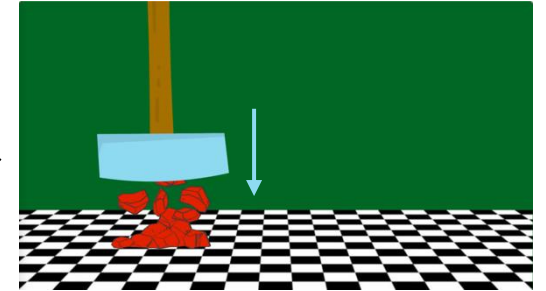
Apply 1<sup>st</sup> function



Intermediate state



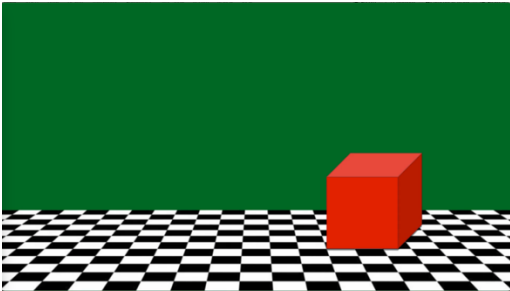
Apply 2<sup>nd</sup> function



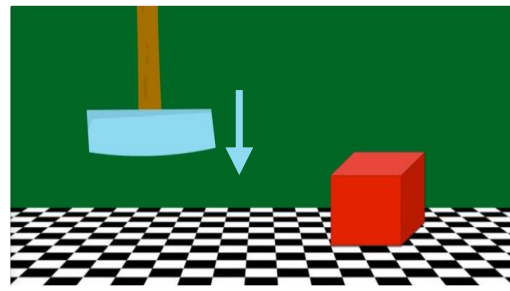
Updated from initial state,  
Matters for outcome of 2<sup>nd</sup> function  
Need to maintain this update

## Smash-First

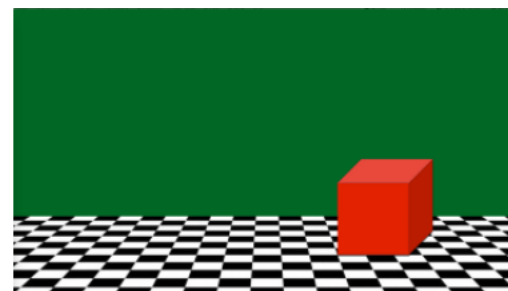
Initial state



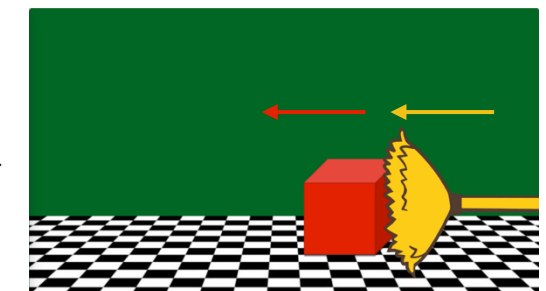
Apply 1<sup>st</sup> function



Intermediate state



Apply 2<sup>nd</sup> function



Same as initial state  
No updating needed  
It's okay to ignore 1<sup>st</sup> function

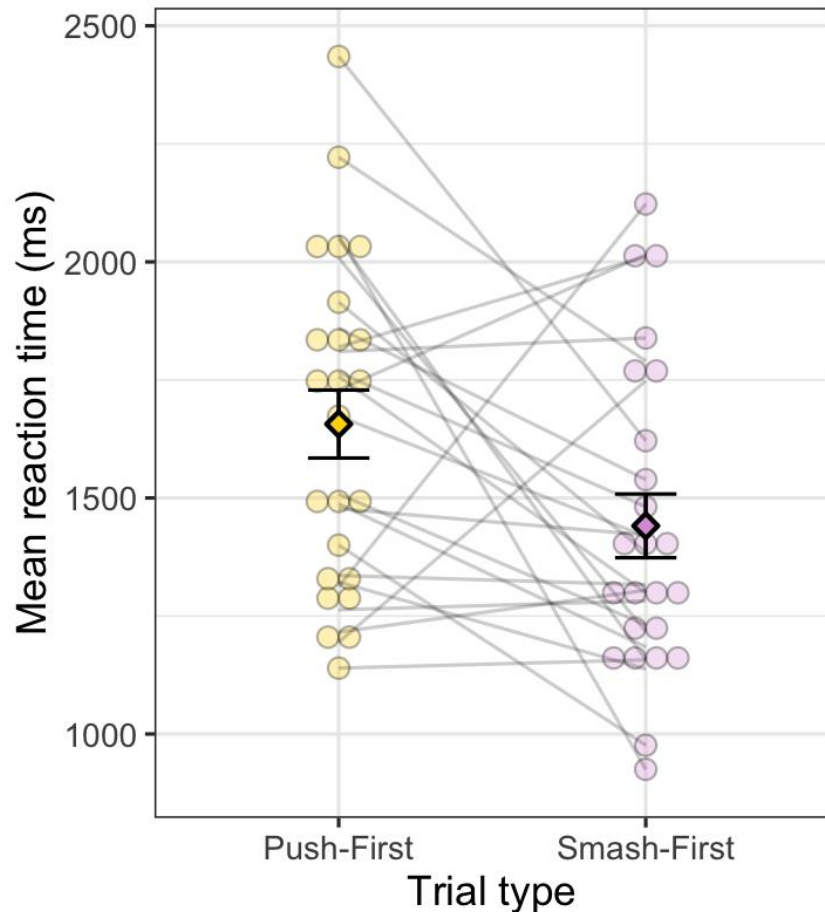
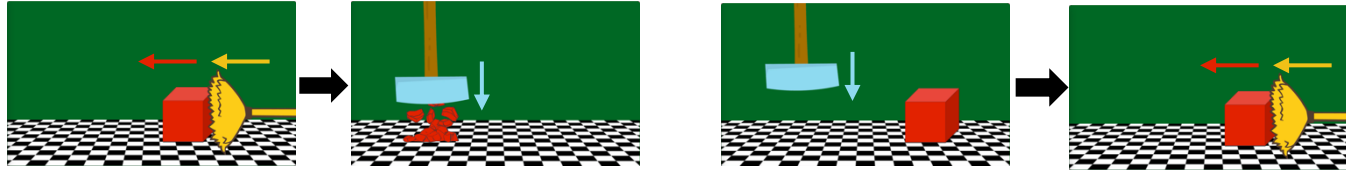
# Is Push-First harder than Smash-First?

- Push-First requires genuine composition: to reason about *smash()*, need to maintain the updated location from the *push()*
- Smash-First
  - Allows a pruning strategy: can ignore *smash()* because it changed nothing
  - Infants can correctly reason about *smash()*, because obj location not updated

Two possibilities, both resulting from WM constraints:

1. Infants can only **attend** to the **last** function (as in Piantadosi et al. 2018)
2. It's **costly** to maintain a location **update** before *smash()* \*and\* reason about its effect on *smash()*

# Teasing apart “attend-last” vs. “costly update”



1. Evidence from adult control study

- Adults were slower at processing **Push-First** compared to **Smash-First**
- Supporting “costly update”, even for adults



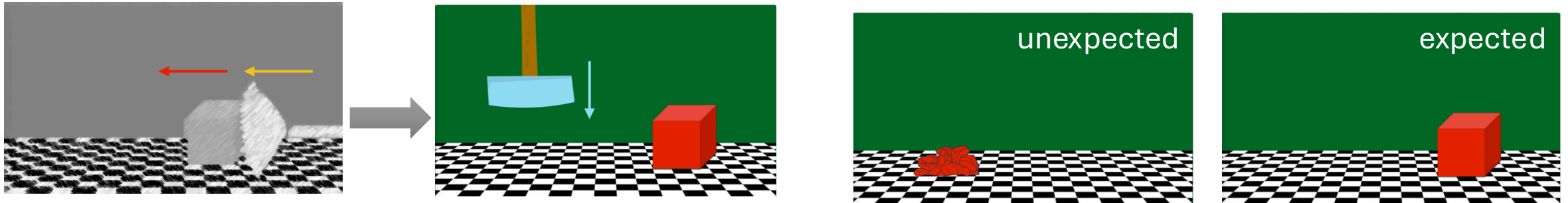


# Teasing apart “attend-last” vs. costly update

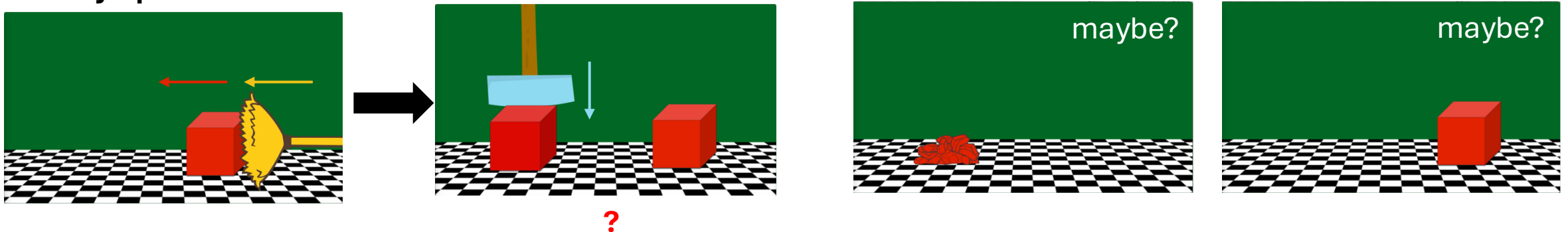
## 2. Test infants on other outcomes/conditions

### 2.1 Other outcomes under Push-First/Feeding

Attend-last:



Costly update

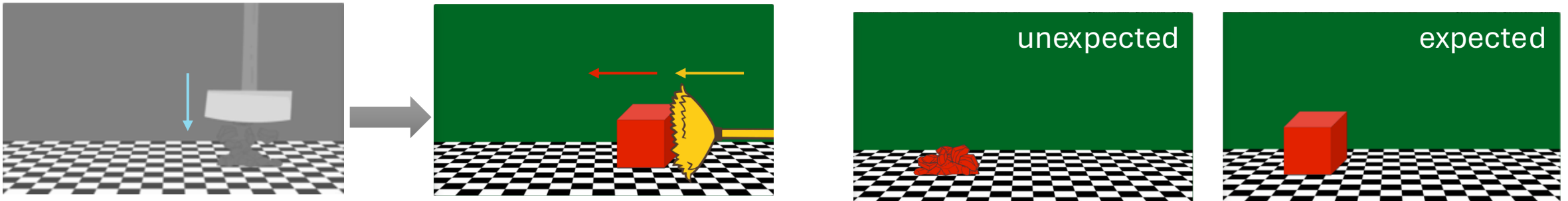


# Teasing apart “attend-last” vs. costly update

## 2. Test infants on other outcomes/conditions

2.2 Other conditions, e.g., object on smasher side, *smash()*, then *push()*

**Attend-last:**



**Costly update**



# Next steps: reduce WM load?

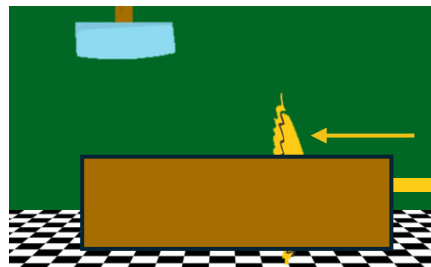
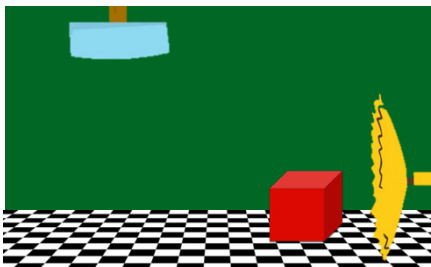
1. Test older infants, e.g., mean = 15mos (Pomiechowska et al. 2024 - CDS)

2. Reduce uncertainty about the occluder

- Add single-function training trials behind the occluder
- “look, the physics stays the same behind the occluder”

3. Reduce uncertainty about the location

- Visible tools at rest, visible broom during *push()*



# Why did 9–11-mos fail at Push-First but succeed in Smash-First?

1. Baseline preference: Smashed is more salient than Intact

No baseline preference (Exp 3)

2. Outcome bias: Unexpected effect (smashed when shouldn't) is more salient than unexpected null effect (intact when shouldn't)

Test infants on single *smash()* functions

3. Working memory constraints: Push-First may be harder than Smash-First

Reduce WM load, test infants on other conditions/outcomes