

ACT-R Software Updates

2021 ACT-R Workshop

Dan Bothell

Carnegie Mellon University

July 12, 2021

Overview

- Issue and bug fixes
- Additions
- Changes
 - Production Compilation
- Performance
- Changes cont.
- Docker

SBCL issue with macOS

- The locking delay is related to the top-level holding a lock
 - Removed a layer of threading for handling commands

Last Year:

* (time (paired-task 2 3))

Evaluation took:

10.324 seconds of real time

1.55% CPU

37,166,382,194 processor cycles

720,880 bytes consed

```
>>> timeit.timeit(stmt='paired.task(2,3)')
```

```
23.807846296000008
```

Now:

* (time (paired-task 2 3))

Evaluation took:

0.010 seconds of real time

100.00% CPU

37,431,836 processor cycles

3,056,432 bytes consed

```
>>> timeit.timeit(stmt='paired.task(2,3)')
```

```
0.2689843200000013
```

- May still encounter some delays with remote interactions
 - When **running is initiated on Lisp side** and external code monitors an action generated during the run

OS related issues for CCL

- Standalone application still runs under macOS 11
 - Don't have a machine with Apple silicon to test yet, but no issues reported
- Had to rebuild standalone application for Windows 10
 - Just after last year's talk

Bug fixes

- Using both `:v` and `:cmdt` to save output to the same file
 - Works as expected instead of bad interleaving/overwriting
- Multiple productions with multiple undefined slots
 - Didn't always reset properly leading to possible mismatching after reset
- Fixed a problem that broke production matching with `:use-tree t`
 - `:use-tree` also works when `:ppm` is enabled now
 - Other updates to make it possibly more useful for smaller models

Additions

- Associative learning that John described at last Workshop
 - extras/associative-learning
- Retrieval requests have a new request parameter :rt-value
 - Adjust the threshold for only this retrieval
 - Like the :mp-value request parameter
- Tutorial models use the new visible-virtuals-available? command
 - Avoid trying to get real response from a virtual window
- Added Import Python Module button to Environment
 - Works with actr.py and is effectively same as calling 'import <name>'

New Production Parsing mechanism [7.24]

- To help simplify production authoring adding “strict safety”
 - Similar to strict harvesting
- Automatically adds a “state free” query to the condition for every buffer that has a request in the action without a corresponding query
- Can be turned off for a buffer with the :do-not-query parameter
 - Goal, Retrieval, Temporal, Visual-location are turned off by default
- The tutorial models still include the explicit queries for now

Articulation time minimum [7.25]

- Articulation time for speak and subvocalize actions now has a minimum of :syllable-rate
- To get same results as before can set :char-per-syllable to 1 (instead of the default 3) and :syllable-rate to .05 (instead of the default 0.15)
- Zbrodoff tutorial model had to be changed slightly to fit data as well

Production Compilation [7.22] & [7.23]

- Better handle strict harvesting issues
 - Null modification to keep a chunk around
 - Buffer empty queries that detected a strict harvesting
- Fewer cases that require instantiation for dynamic slots
- Options for allowing indirect actions to be compiled
- General cleanup of the internal code

Compilation with a null modification

(p a	(p b	Now
=goal>	=goal>	(P PRODUCTION0
state nil	state t	"A & B"
=imaginal>	=imaginal>	=GOAL>
==>	==>	STATE NIL
=imaginal>	=goal>	=IMAGINAL>
=goal>	state 1)	==>
state t)		=GOAL>
		STATE 1)

Previously

Buffer IMAGINAL prevents composition of these productions

because strict harvesting is enabled for the buffer and either the first production cleared the buffer or the combined production would leave the buffer with a chunk while the original pair would have left it empty.

Compilation with a buffer empty query

(p a	(p b	Previously	Now
=goal>	=goal>	(P PRODUCTION0	(P PRODUCTION0
state nil	state t	"A & B"	"A & B"
=imaginal>	?imaginal>	=GOAL>	=GOAL>
==>	buffer empty	STATE NIL	STATE NIL
=goal>	==>	=IMAGINAL>	=IMAGINAL>
state t)	=goal>	?IMAGINAL>	==>
	state 1)	BUFFER EMPTY	=GOAL>
		==>	STATE 1)
		=GOAL>	
		STATE 1)	

Compilation with dynamic slots

(p a	(p b	Previously(*)	Now
=goal>	=goal>	(P PRODUCTION0	(P PRODUCTION0
state nil	=s =v	"A & B"	"A & B"
=imaginal>	=imaginal>	=GOAL>	=GOAL>
slot =x		STATE NIL	STATE NIL
value =y	slot =s	=IMAGINAL>	=IMAGINAL>
==>	==>	VALUE =Y	VALUE =Y
=imaginal>	!output! =v	SLOT COLOR	SLOT =S
=goal>)	==>	==>
state t		=GOAL>	=GOAL>
=x =y)		COLOR =Y	=S =Y
		STATE T	STATE T
		!OUTPUT! =Y	!OUTPUT! =Y
))

Compilation of indirect actions

- Still slightly speculative so handled with two new parameters :cia (compile indirect actions) and :rir (retain indirect retrievals) which default to nil

(p a

=goal>

value =v

state nil

=>

=goal>

state t

+retrieval> =v)

Previously (and with :cia nil)

Production Compilation process started for A

Production A is not valid for compilation

because it has an indirect action with the RETRIEVAL buffer

With :cia t

```
(p a
  =goal>
    value =v
    state nil
==>
  =goal>
    state t
  +retrieval> =v)
```

```
(p b
  =goal>
    state t
==>
  =goal>
    state 1)
```

```
(P PRODUCTION0
  "A & B"
  =GOAL>
    STATE NIL
    VALUE BLUE
==>
  =GOAL>
    STATE 1
  +RETRIEVAL>
    COLOR BLUE)
```

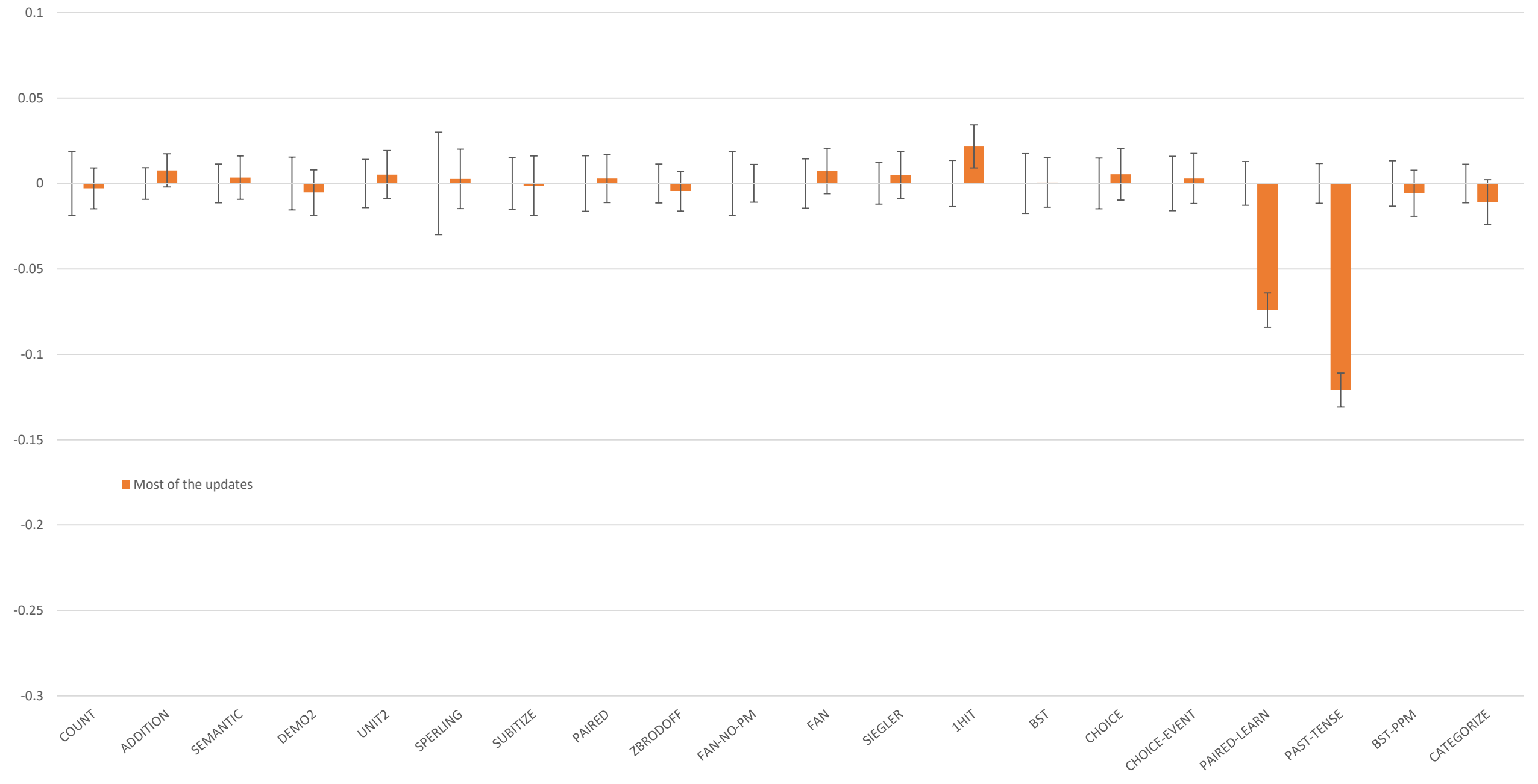
With :cia t and :rir t

```
(p a
  =goal>
    value =v
    state nil
==>
  =goal>
    state t
  +retrieval> =v)
```

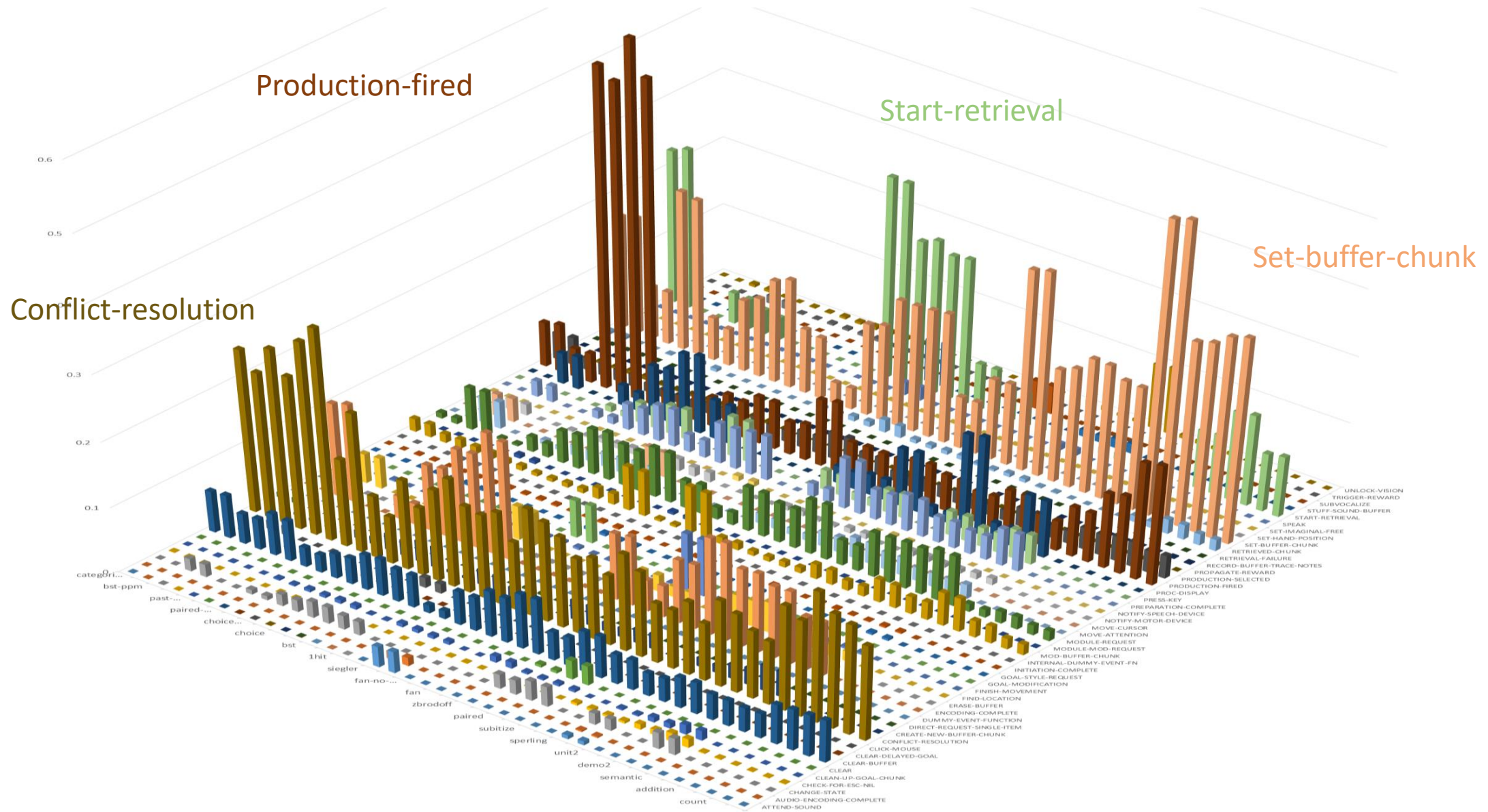
```
(p b
  =goal>
    state t
==>
  =goal>
    state 1)
```

```
(P PRODUCTION0
  "A & B"
  =GOAL>
    STATE NIL
    VALUE =V
==>
  =GOAL>
    STATE 1
  +RETRIEVAL> =V)
```

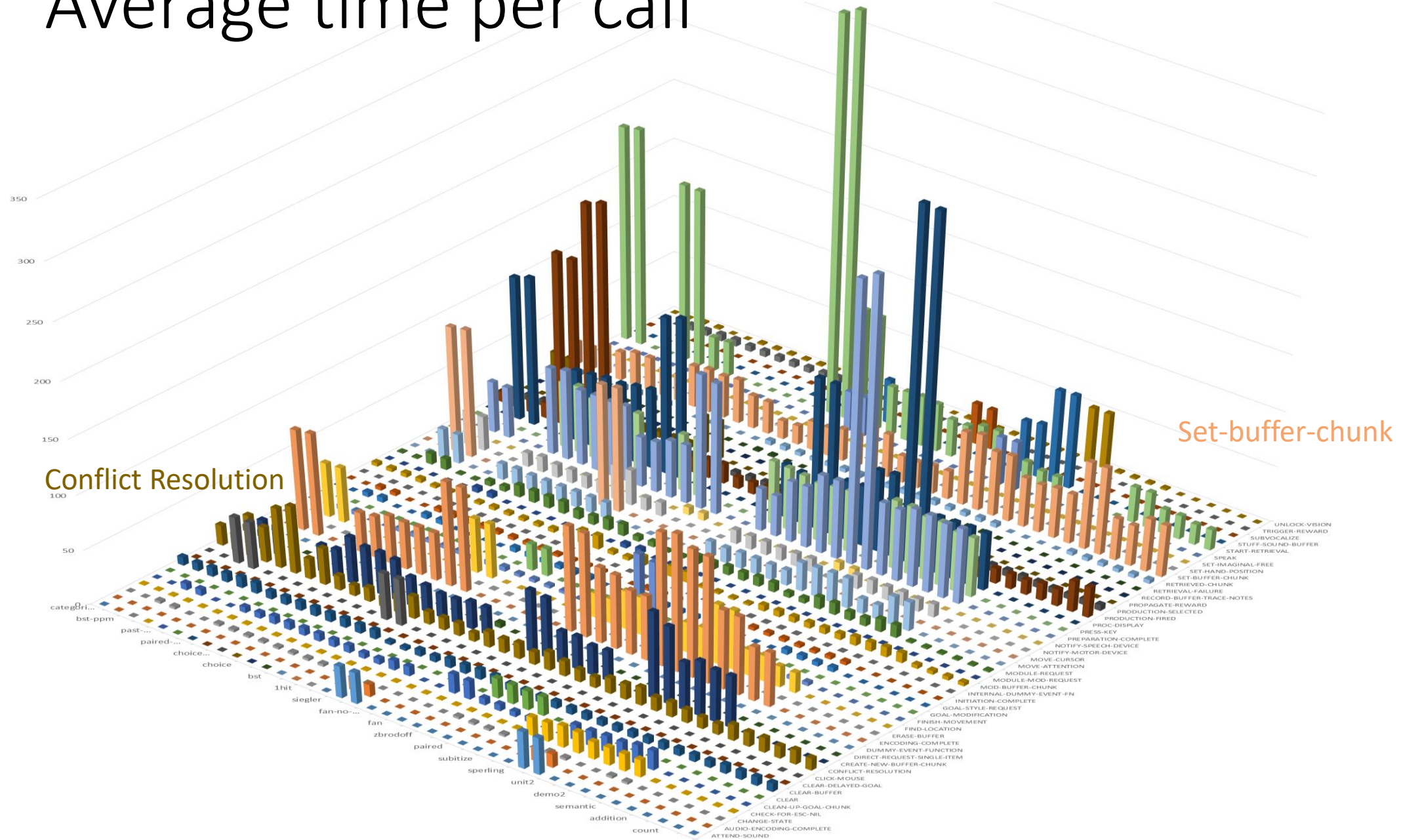
Time Change Proportion



Events' Proportion of Run time



Average time per call



Why?

- Creating the new name for the copy!
 - Create a name (add a new number to the end of the name)
 - Check if it's already used for another chunk
 - Intern the new symbol

Proposed Improvement

- Each buffer has a fixed chunk associated with it
 - Always the same name – only generated once
- Setting the buffer just requires updating that chunk

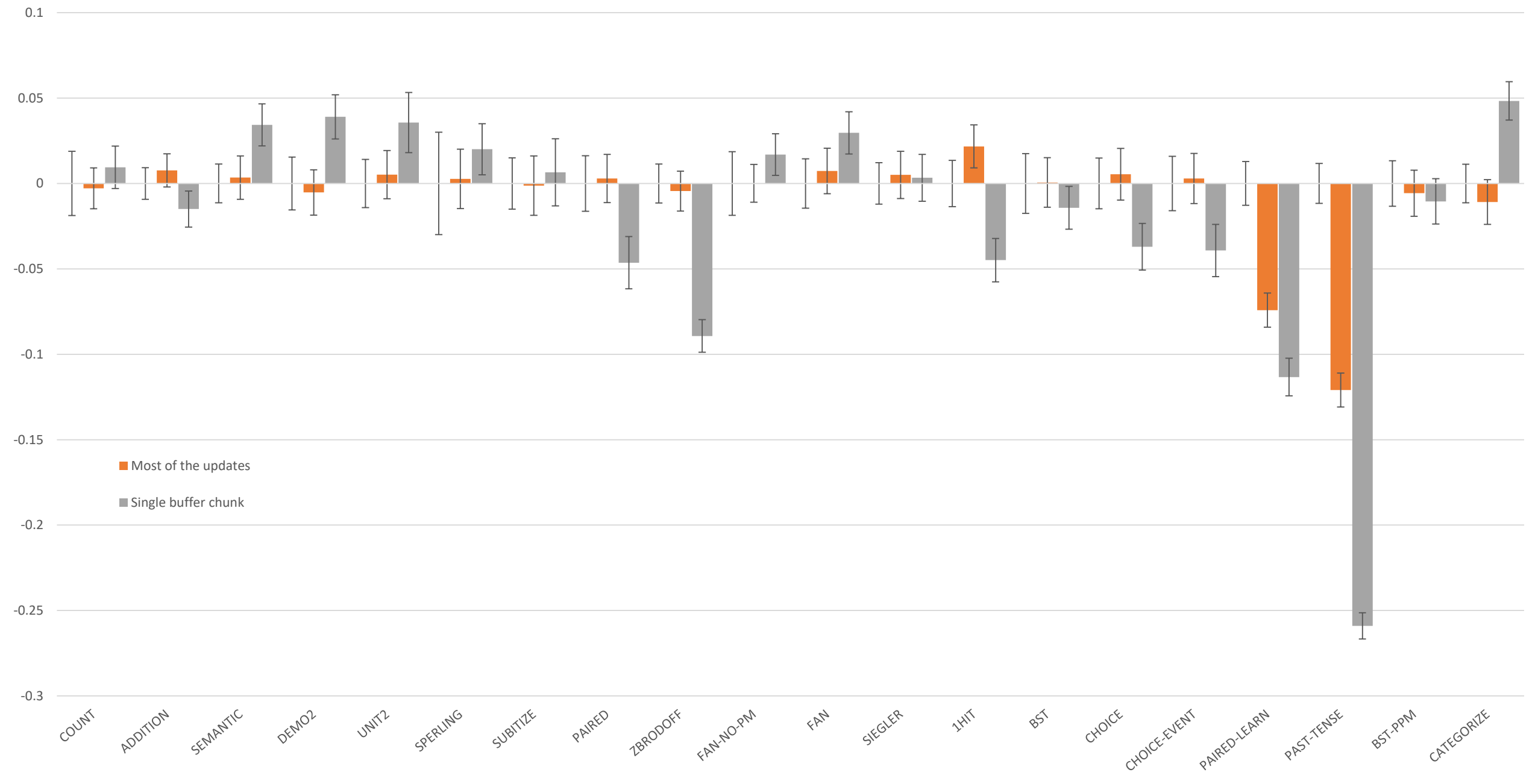
Possible Issues

- What if something needs unique names for the chunks?
 - Using the production buffer variables
 - Linking the chunks in visual-location & visual and aural-location & aural
 - Storing the chunk in DM

Addressing those issues

- Provide a command to force unique names – buffer-requires-copies
 - Procedural module can set it if a production uses the buffer variable
 - Perceptual modules always set it for the *-location buffers
- DM can copy it when necessary
 - Merging can happen without a unique name
- Tested with tutorial models
 - No changes necessary to produce same results
 - There are differences in the traces
- Changes typically only necessary at the module level
 - Possibly in code that runs a model if it depends on specific chunk names
 - The chunk-not-storable command can be used to check a chunk before recording it

Time Change Proportion



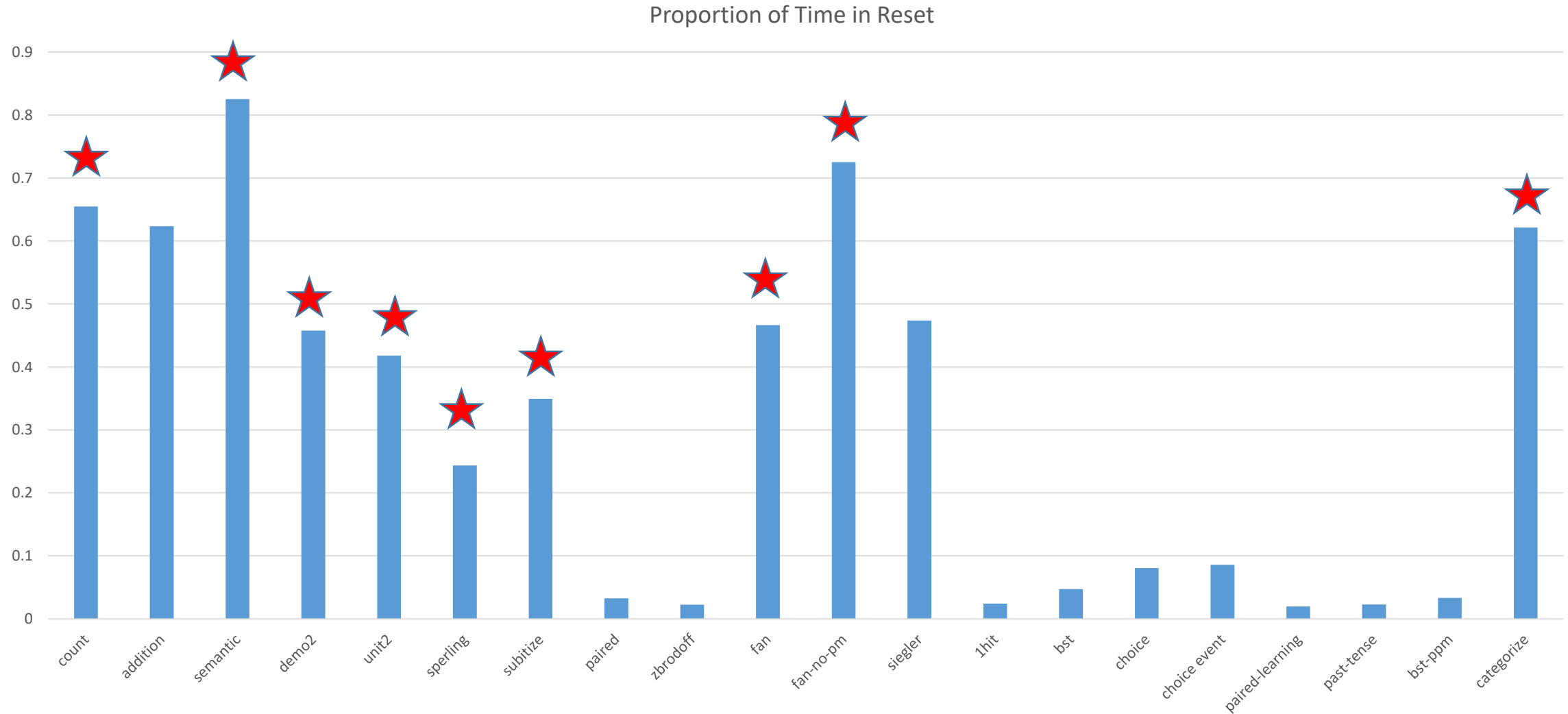
Set-buffer-chunk

- Set-buffer-chunk can now be passed the description of the chunk instead of needing to create a chunk in advance
 - Either a list of slot-value pairs or a chunk-spec
- Original past tense model run 20000 trials generates 143450 chunks
- Running as-is with the update generates 60084 chunks
- Changed the experiment code so that it passes a description of the chunk to set-buffer-chunk instead of explicitly generating one for the practice items
 - Only generates 84 chunks! (~44% faster than last year)

Code changes are now committed [7.26]

- Documentation updates are still ongoing

What about the models that got slower



Docker

- Version of ACT-R running under SBCL in a container available
- Also includes
 - Jupyter notebook server and pages to run tutorial tasks
 - Nodejs server for the HTML version of ACT-R Environment tools
- Instructions on the ACT-R website software page
http://act-r.psy.cmu.edu/actr7.x/container_readme.txt
- Still rough around the edges
 - Startup scripts in the container are fragile
 - No content in the Jupyter notebooks other than running the task functions
- Thanks to Andrea Stocco and Don Morrison

Running the container online

- Can run it with a Lisp interface using the play-with-docker site
 - Virtual linux machine in a browser available free with a Docker account
 - Limits on memory usage and time

<https://labs.play-with-docker.com/>

Commands needed once session running:

```
docker pull db30/act-r-container
```

```
docker run -i -p 4000:4000 db30/act-r-container act-r.sh
```

Binder

- If the Dockerfile and Jupyter files are available from a public repository Binder can be used to run it online from either a public or private server

<https://github.com/db30/act-r-container>

- Can run the ACT-R materials on the free mybinder server

- Limited simultaneous users, memory usage, and time

<https://mybinder.org/v2/gh/db30/act-r-container/main?filepath=tutorial.ipynb>

Future

- Update and extend the docker/Jupyter materials?