

## Incorporating Feature-Model Edits with Incremental d-DNNF Compilation

FOSD'25 | Chico Sundermann, Heiko Raab, Thomas Thüm | 28.03.2025

# Motivation Product Lines



Your Selection

18-inch 5-spoke alloy wheels  
without extra charge



# Motivation Product Lines



Your Selection  
18-inch 5-spoke alloy wheels  
without extra charge

## PC BUILDER

GET COMPATIBLE RECOMMENDATIONS  
PICK YOUR IDEAL CORSAIR COMPONENTS



# Motivation Product Lines

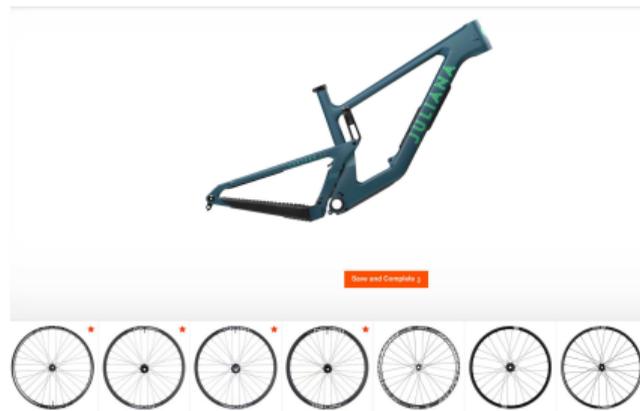


Your Selection  
18-inch 5-spoke alloy wheels  
without extra charge



## PC BUILDER

GET COMPATIBLE RECOMMENDATIONS  
PICK YOUR IDEAL CORSAIR COMPONENTS



# Motivation Product Lines

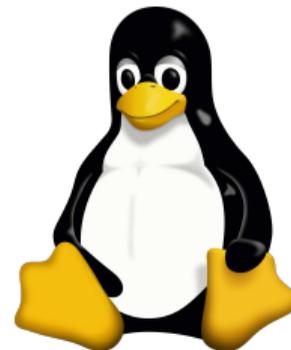
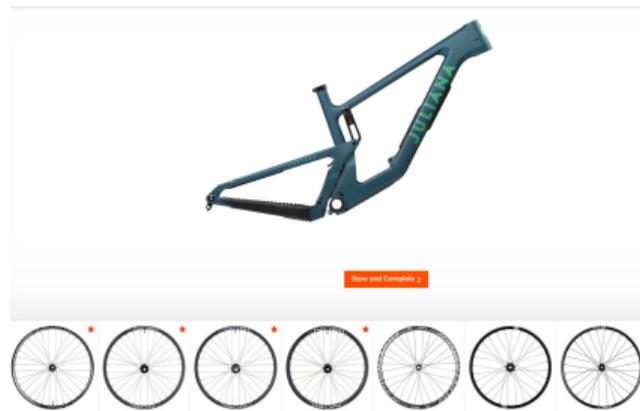


Your Selection  
18-inch 5-hole-spoke alloy wheels  
without extra charge



## PC BUILDER

GET COMPATIBLE RECOMMENDATIONS  
PICK YOUR IDEAL CORSAIR COMPONENTS



# Motivation Feature Dependencies

 **Compatibility:** Warning! These parts have potential issues or incompatibilities. See [details](#) below.

**GROUPSET** ⓘ

-  Shimano GRX600 2x12sp
-  Shimano 105 – 2x12
-  Sram Apex XPLR AXS 1x12
-  Shimano GRX800 2x12sp
-  SRAM Rival AXS XPLR – 1x12
-  Shimano 105 Di2 Disc 2x12
-  Shimano Ultegra Di2 - 2x12

```
config SECURITY_INFINIBAND
    bool "Infiniband Security Hooks"
    depends on SECURITY && INFINIBAND
```

 **ASRock B650M Pro RS WiFi Micro ATX AM5**

**CPU Sockets**

- CPU\_1 (AM5)

**Memory Slots**

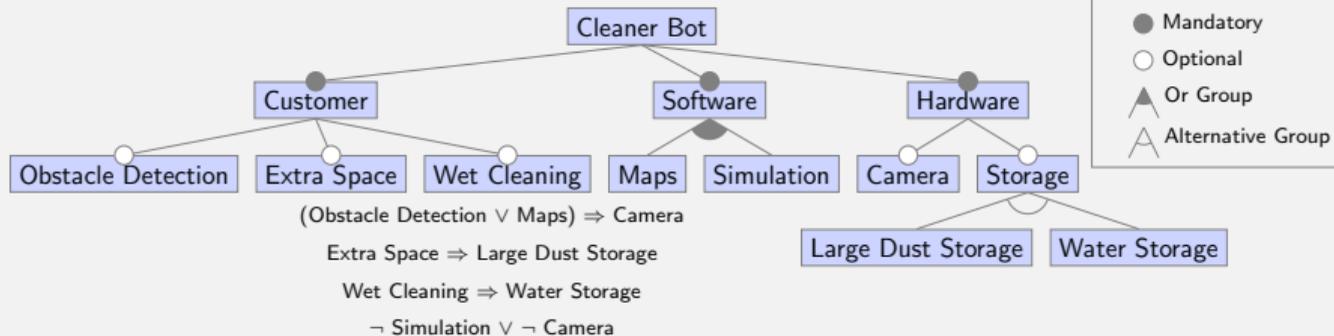
- RAM\_1 (288-pin DIMM)
- RAM\_2 (288-pin DIMM)
- RAM\_3 (288-pin DIMM)
- RAM\_4 (288-pin DIMM)

# Motivation Feature Dependencies

⚠ **Compatibility:** Warning! These parts have potential issues or incompatibilities. See [details](#) below.

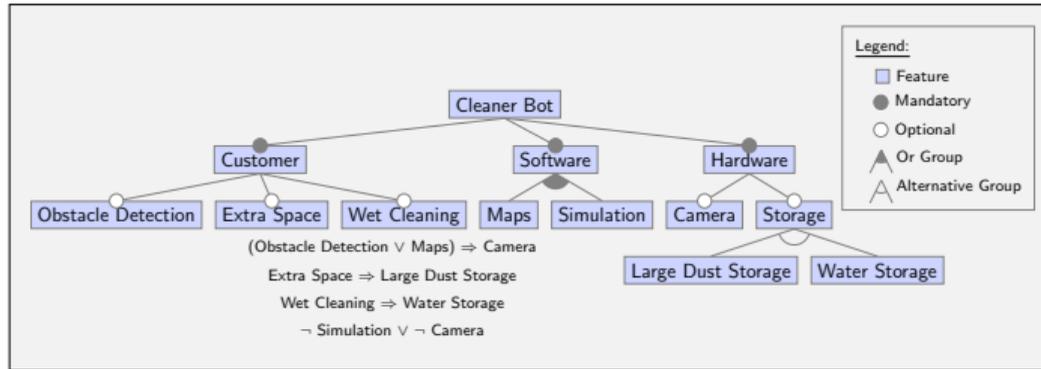
GROUPSET ⓘ

X AM5

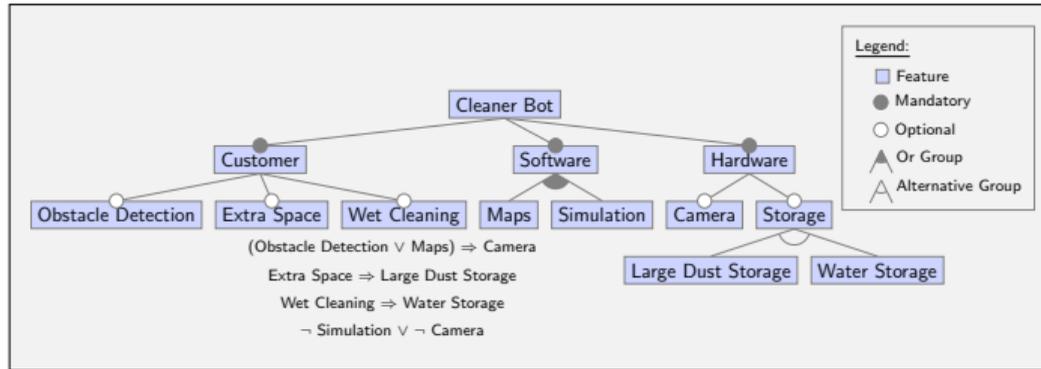


Shimano Ultegra Di2 - 2x12

# Motivation Plenty Analyses

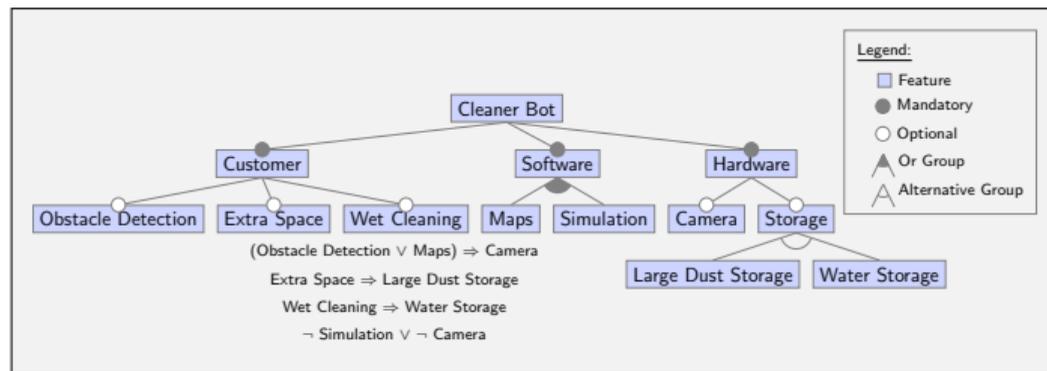


# Motivation Plenty Analyses



Can we configure a robot?

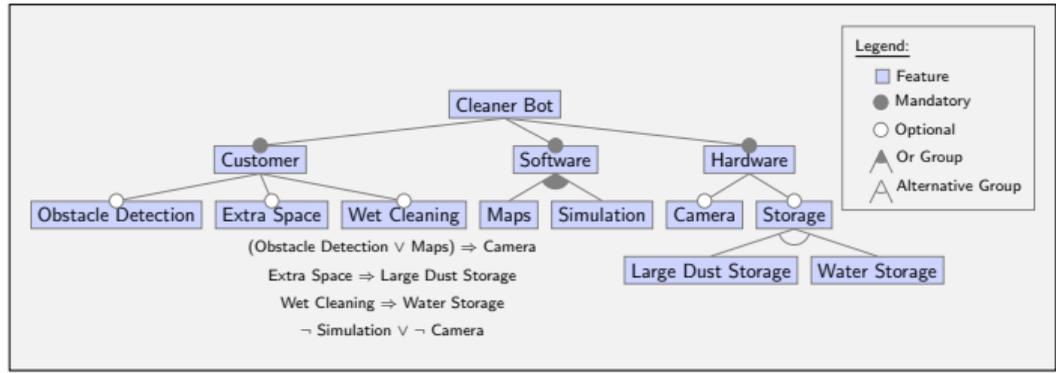
# Motivation Plenty Analyses



Can we configure a robot?

Is there a robot with camera?

# Motivation Plenty Analyses

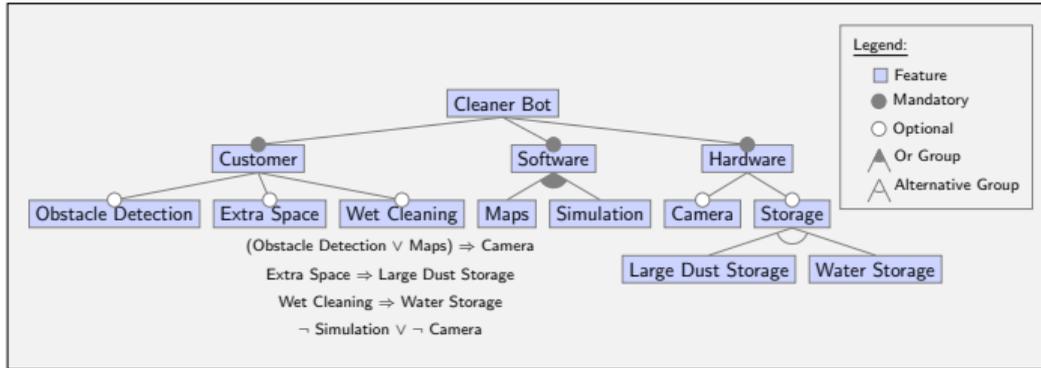


Can we configure a robot?

Is there a robot with camera?

How many robots?

# Motivation Plenty Analyses



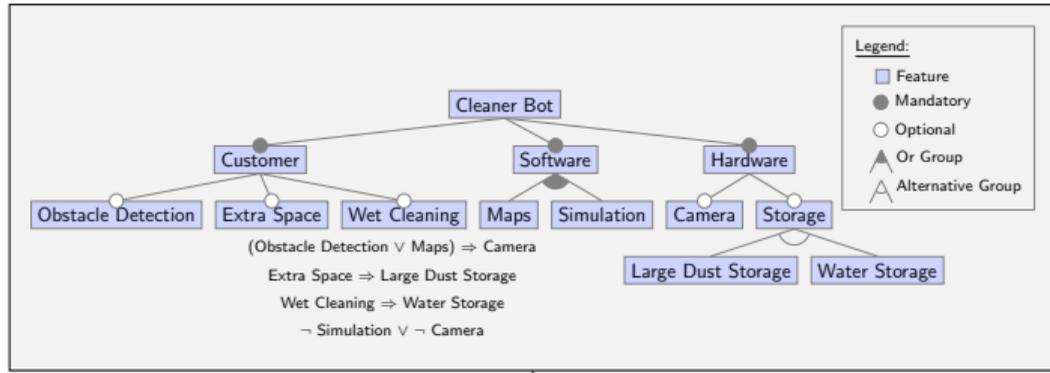
Can we configure a robot?

Is there a robot with camera?

How many robots?

Which robots to test?

# Motivation Plenty Analyses



Can we configure a robot?

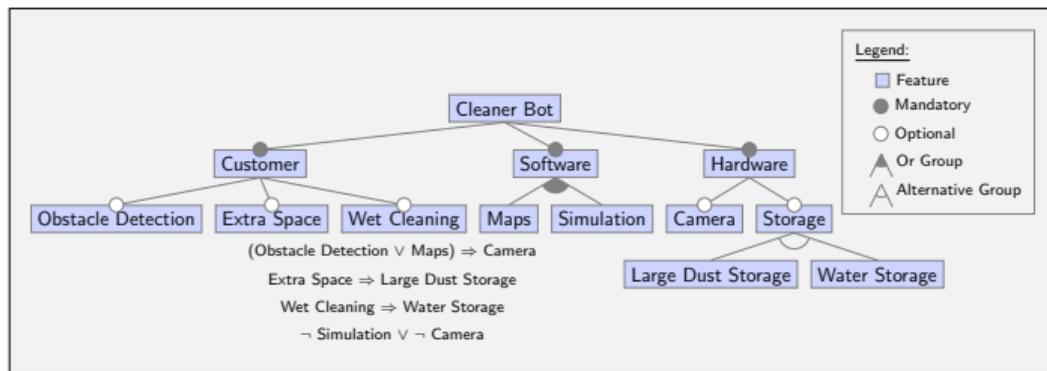
Is there a robot with camera?

How many robots?

Which robots to test?

$Customer \wedge Software \wedge (Maps \vee Simulation) \wedge \dots$

# Motivation Plenty Analyses



Can we configure a robot?

Is there a robot with camera?

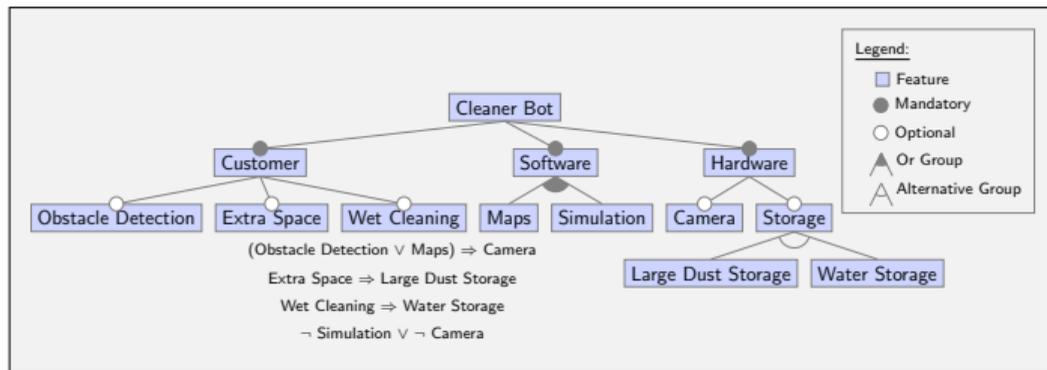
How many robots?

Which robots to test?

Customer  $\wedge$  Software  $\wedge$  (Maps  $\vee$  Simulation)  $\wedge$  ...



# Motivation Plenty Analyses



Can we configure a robot?

Is there a robot with camera?

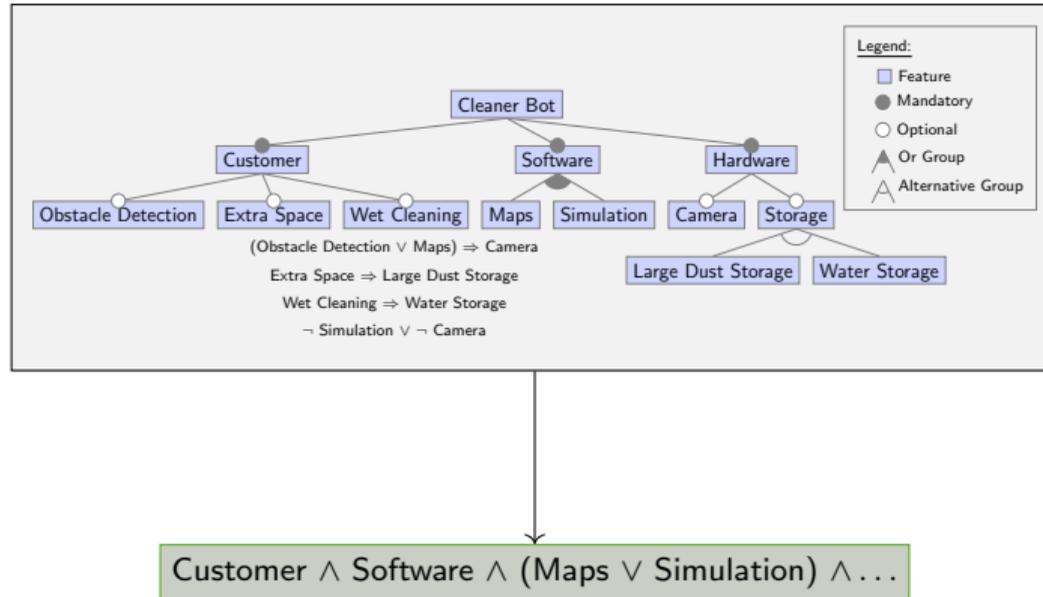
How many robots?

Which robots to test?

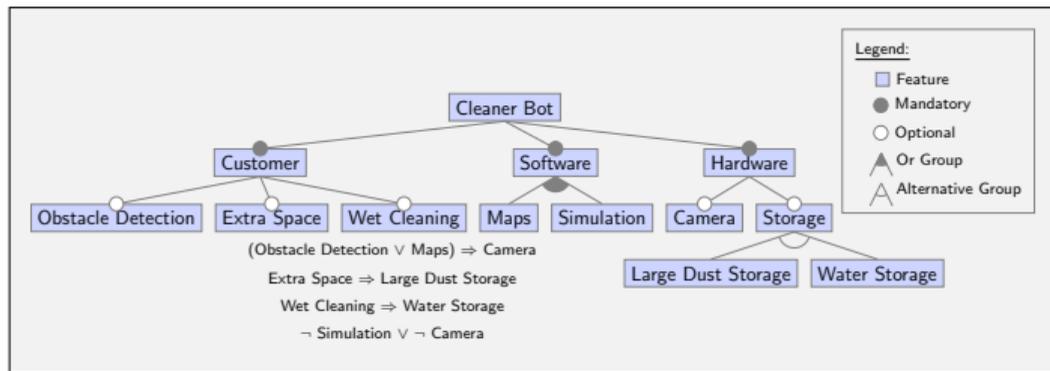
Customer  $\wedge$  Software  $\wedge$  (Maps  $\vee$  Simulation)  $\wedge$  ...



# Motivation Knowledge Compilation

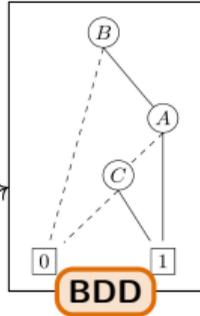


# Motivation Knowledge Compilation

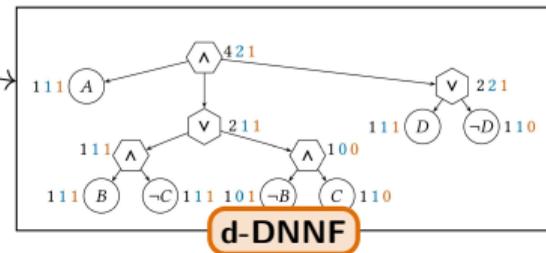


Customer  $\wedge$  Software  $\wedge$  (Maps  $\vee$  Simulation)  $\wedge$  ...

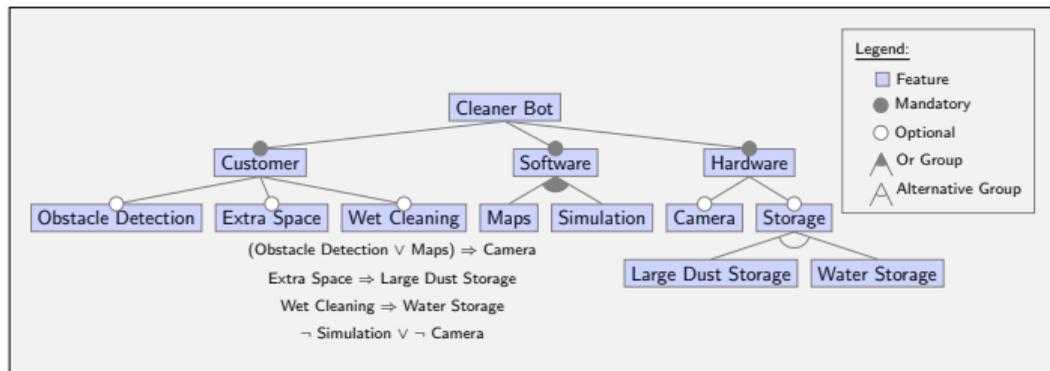
Compile



Compile



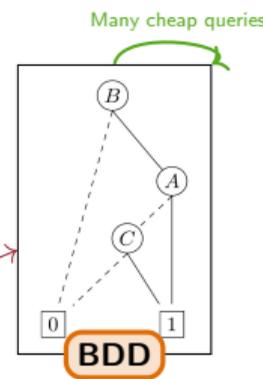
# Motivation Knowledge Compilation



Customer  $\wedge$  Software  $\wedge$  (Maps  $\vee$  Simulation)  $\wedge$  ...

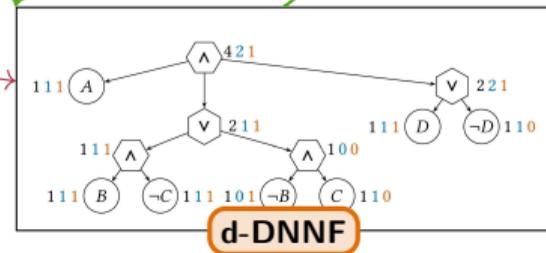
One-time expensive

Many cheap queries

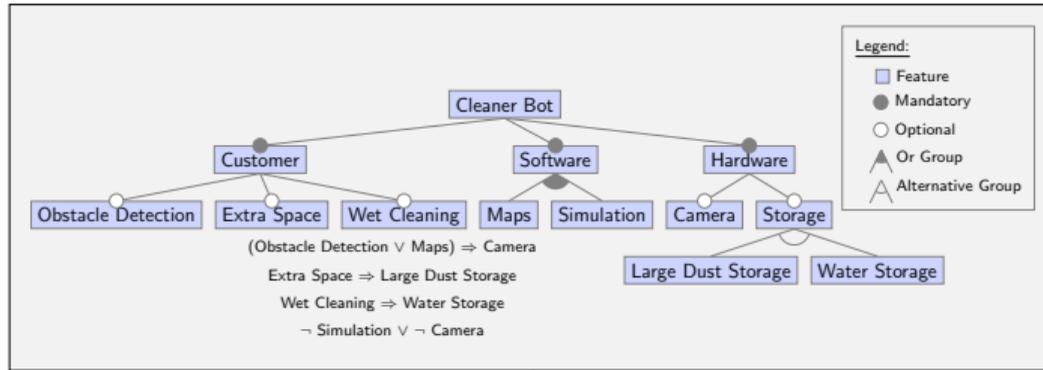


One-time expensive

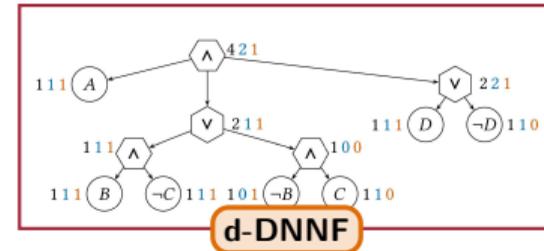
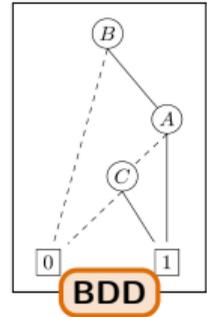
Many cheap queries



# Motivation Knowledge Compilation



Customer  $\wedge$  Software  $\wedge$  (Maps  $\vee$  Simulation)  $\wedge$  ...



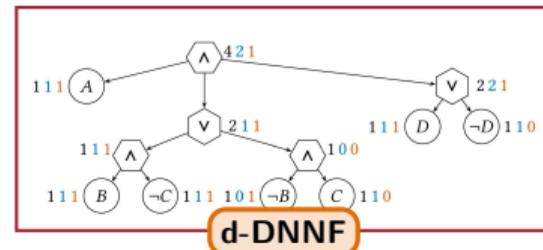
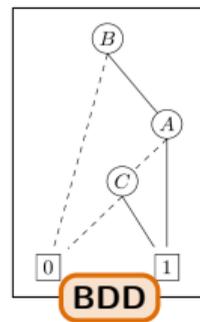
# Motivation Knowledge Compilation

Annals of Mathematics and Artificial Intelligence  
<https://doi.org/10.1007/s10472-023-09906-6>

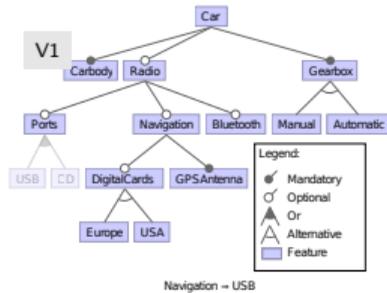
Check for updates

**On the benefits of knowledge compilation for feature-model analyses**

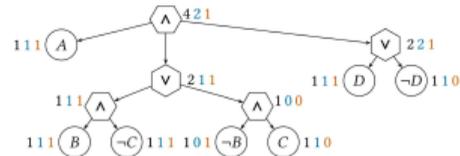
Chico Sundermann<sup>1</sup> · Elias Kuitert<sup>2</sup> · Tobias Heß<sup>1</sup> · Heiko Raab<sup>1</sup> · Sebastian Krieter<sup>1</sup> · Thomas Thüm<sup>1</sup>



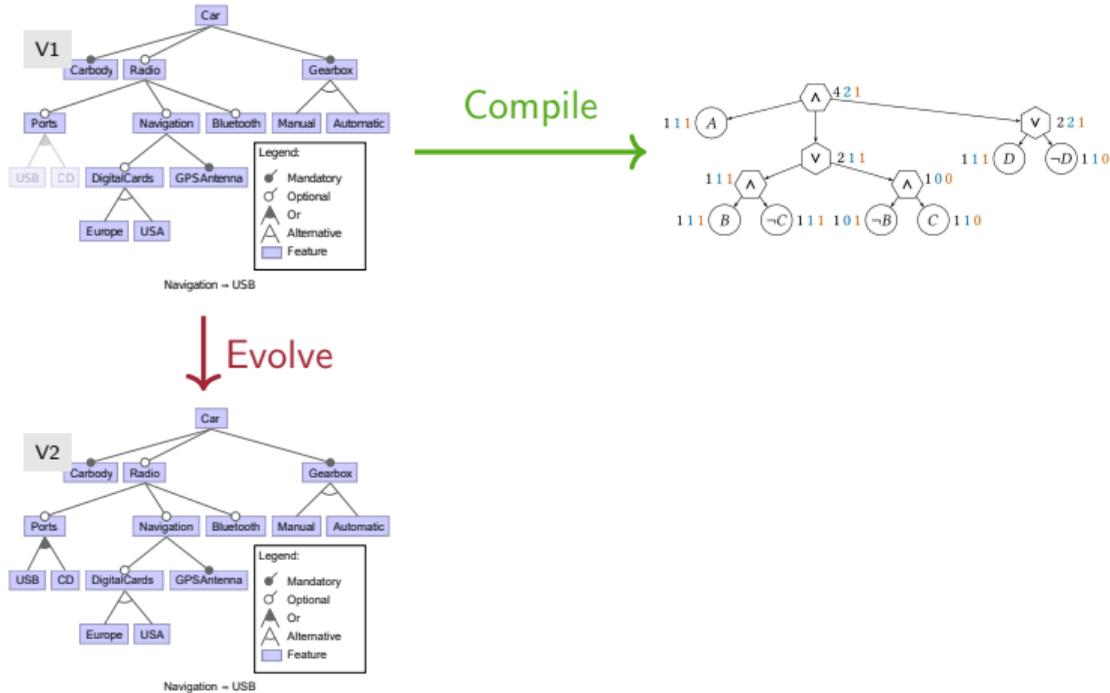
# Incremental d-DNNF Compilation



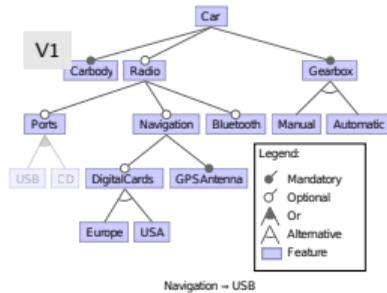
Compile →



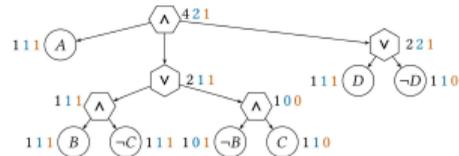
# Incremental d-DNNF Compilation



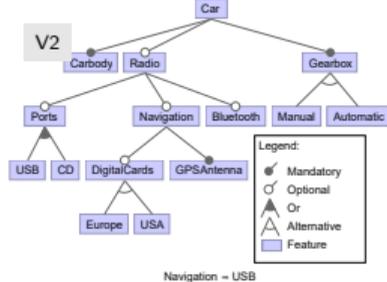
# Incremental d-DNNF Compilation



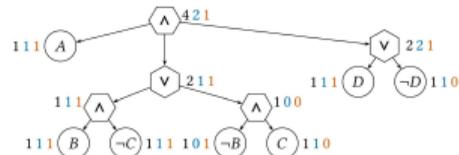
Compile



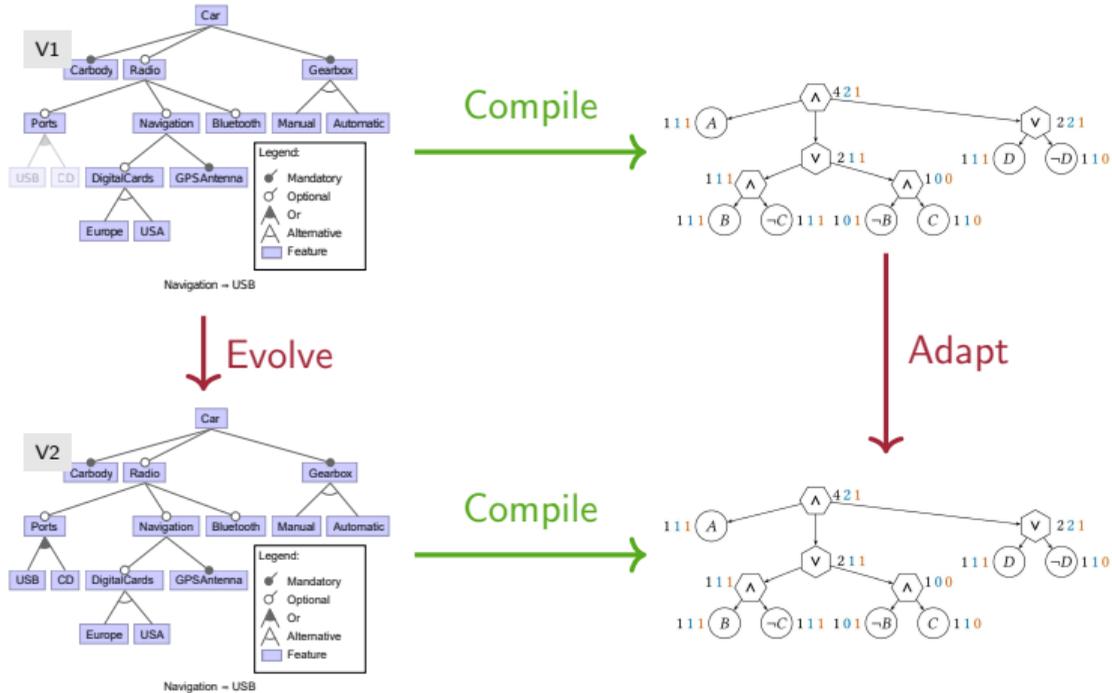
Evolve



Compile



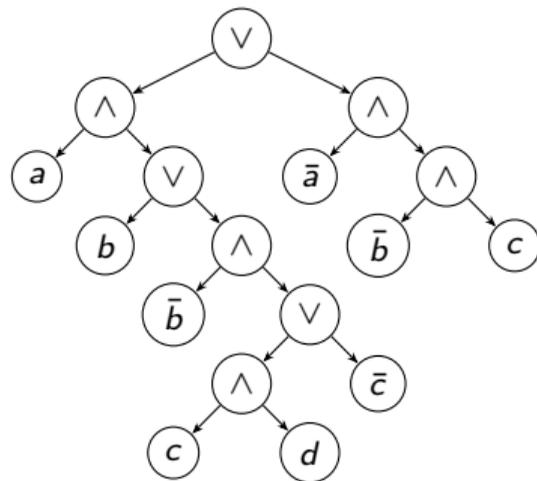
# Incremental d-DNNF Compilation



## Approach Identify Part Requiring Update

- Recurse d-DNNF to find parts sharing variables with clause
- Apply assumptions while recursing to skip parts unchanged by clause

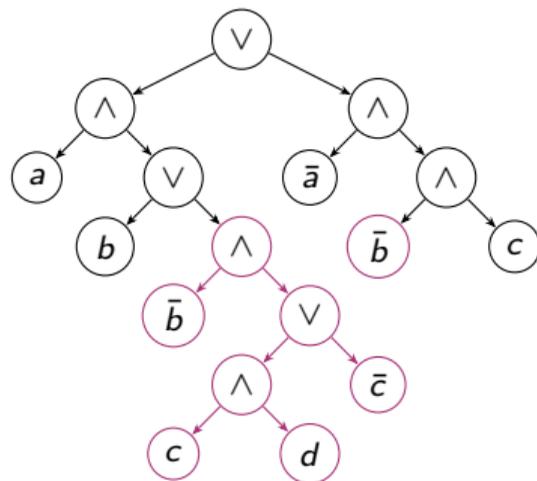
Change:  $b \vee d$



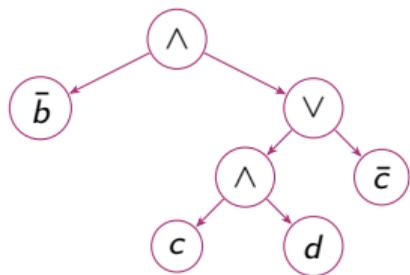
## Approach Identify Part Requiring Update

- Recurse d-DNNF to find parts sharing variables with clause
- Apply assumptions while recursing to skip parts unchanged by clause

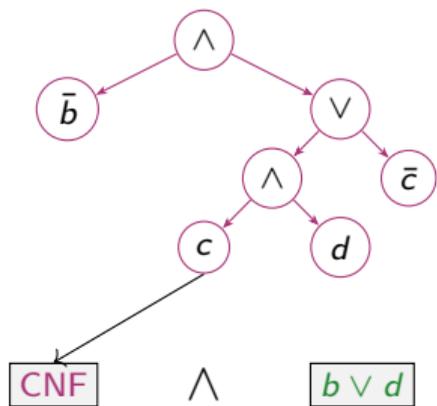
Change:  $b \vee d$



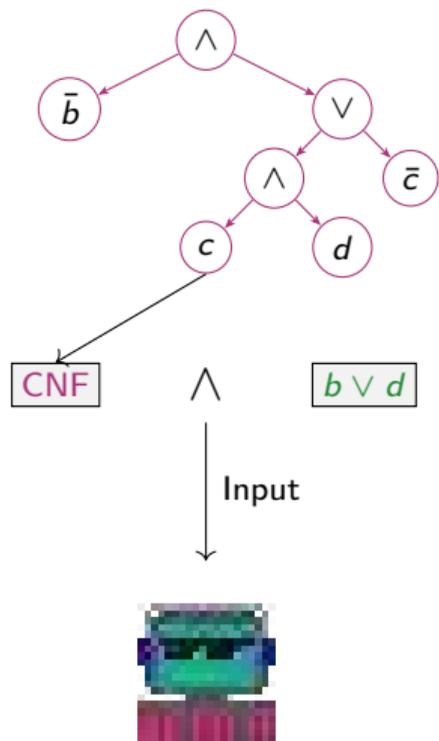
## Approach Partially Recompile & Reattach



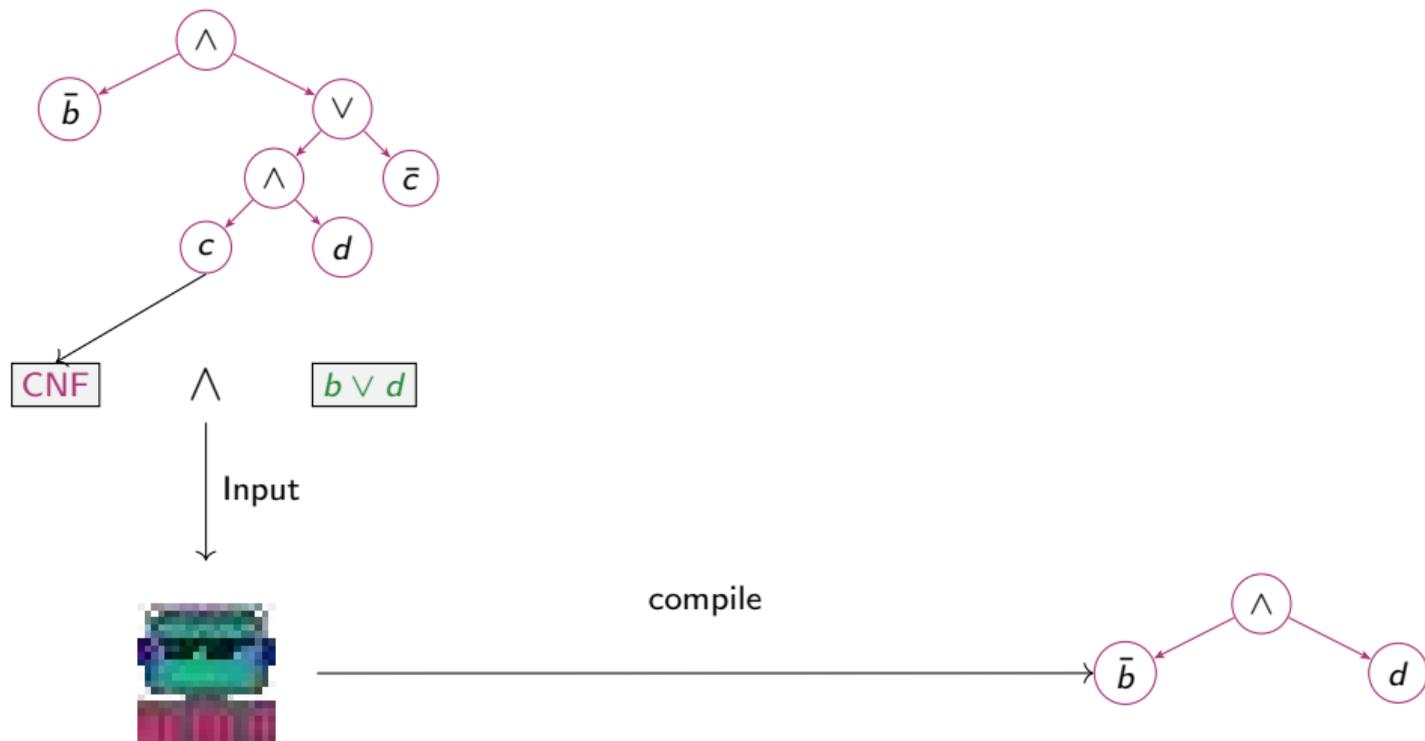
## Approach Partially Recompile & Reattach



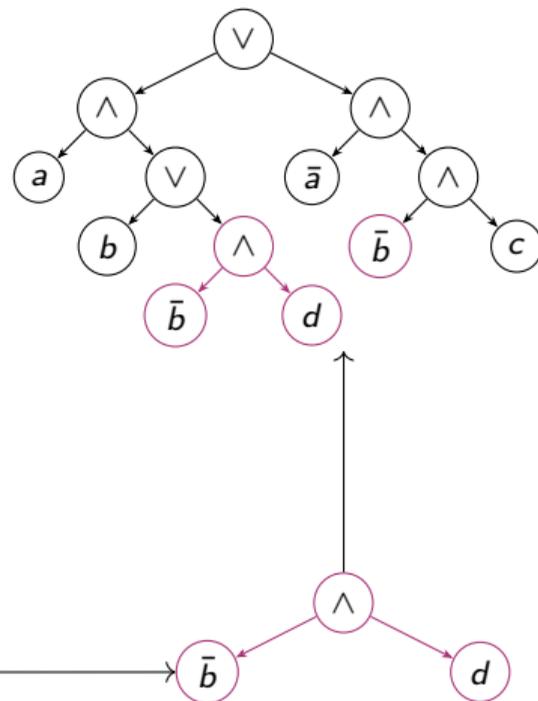
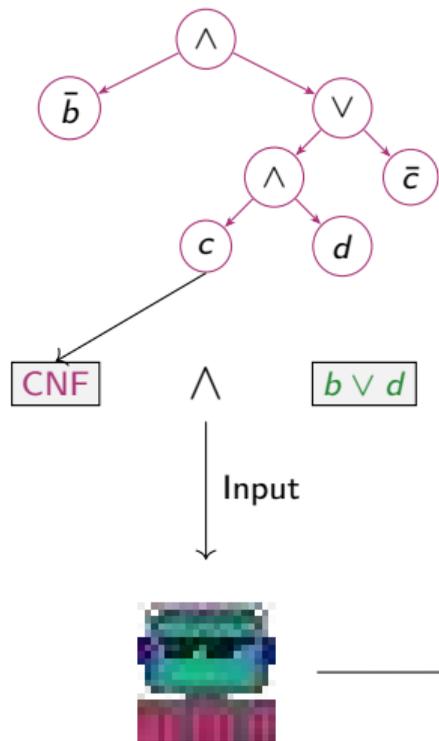
# Approach Partially Recompile & Reattach



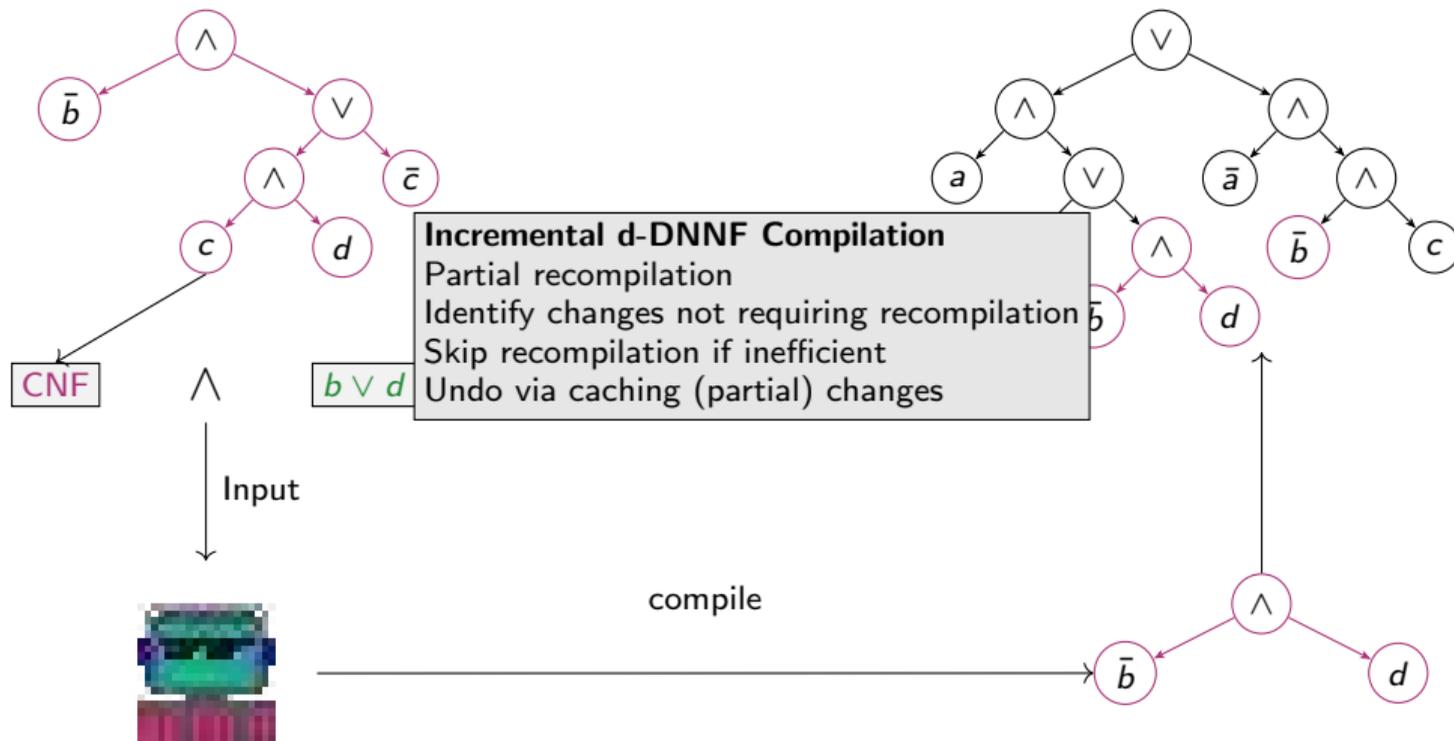
# Approach Partially Recompile & Reattach



# Approach Partially Recompile & Reattach



# Approach Partially Recompile & Reattach



# Evaluation Experiment Design

- Goal: Examine performance benefits

Domain	Models	Features	Clauses	Cardinality
automotive	2	2,513 – 18,616	10,311 – 350,149	$5.3^{210} - 1.7^{1534}$
business	1	1,920	61,362	$2.0^{397}$
cloud	1	106	156	$5.4^{20}$
database	1	117	417	$3.2^1$
deep-learning	2	3,296 – 6,867	8,004 – 17,431	$1.3^{185} - 2.8^{630}$
e-commerce	2	287 – 2,238	421 – 2,615	$2.3^{49} - 8.0^{558}$
finance	3	176 – 771	280 – 7,238	$9.7^{13} - 5.3^{31}$
games	1	144	769	$4.2^{18}$
hardware	2	172 – 377	309 – 1,356	$3.3^{21} - 2.3^{27}$
navigation	2	103 – 145	185 – 378	$3.2^{10} - 1.8^{12}$
security	1	158	457	$1.2^{14}$
systems software	14	96 – 1,580	183 – 15,692	$8.3^{11} - 4.1^{409}$
text	1	137	179	$1.3^{23}$

# Evaluation Experiment Design

- Goal: Examine performance benefits
- Subject systems
  - Industrial feature Models
  - Various domains, sizes

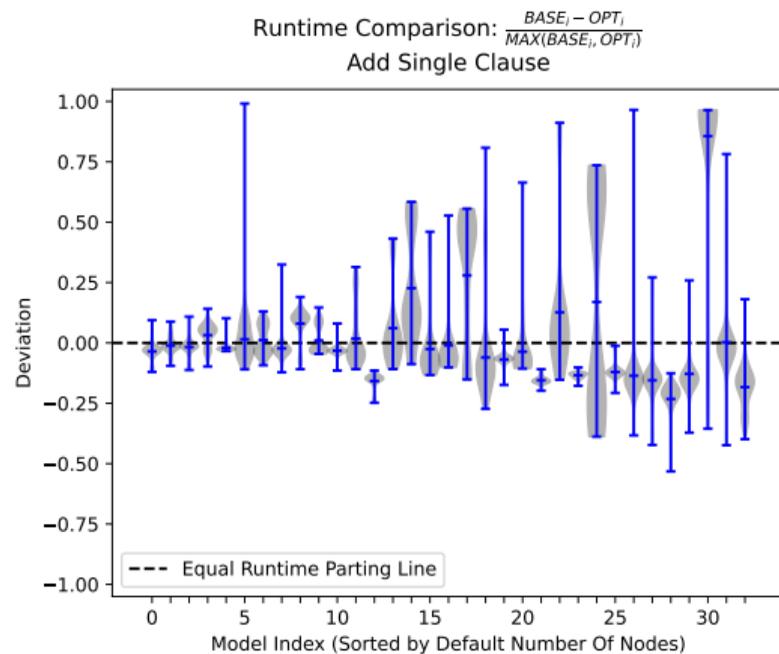
Domain	Models	Features	Clauses	Cardinality
automotive	2	2,513 – 18,616	10,311 – 350,149	$5.3^{210} - 1.7^{1534}$
business	1	1,920	61,362	$2.0^{397}$
cloud	1	106	156	$5.4^{20}$
database	1	117	417	$3.2^1$
deep-learning	2	3,296 – 6,867	8,004 – 17,431	$1.3^{185} - 2.8^{630}$
e-commerce	2	287 – 2,238	421 – 2,615	$2.3^{49} - 8.0^{558}$
finance	3	176 – 771	280 – 7,238	$9.7^{13} - 5.3^{31}$
games	1	144	769	$4.2^{18}$
hardware	2	172 – 377	309 – 1,356	$3.3^{21} - 2.3^{27}$
navigation	2	103 – 145	185 – 378	$3.2^{10} - 1.8^{12}$
security	1	158	457	$1.2^{14}$
systems software	14	96 – 1,580	183 – 15,692	$8.3^{11} - 4.1^{409}$
text	1	137	179	$1.3^{23}$

# Evaluation Experiment Design

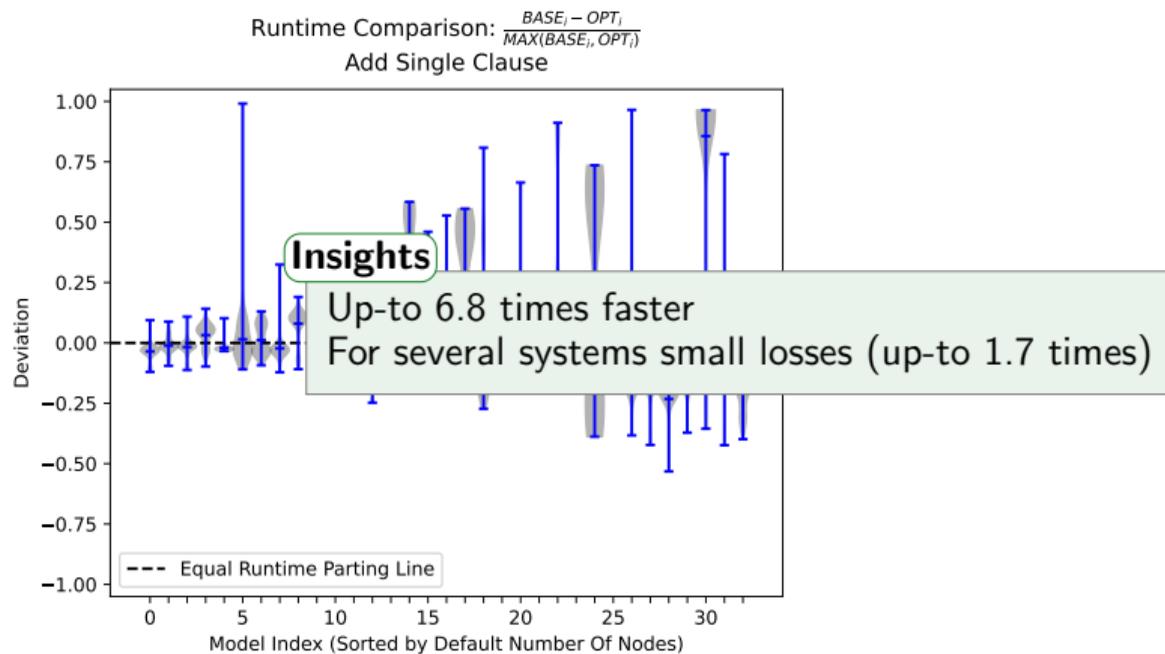
- Goal: Examine performance benefits
- Subject systems
  - Industrial feature Models
  - Various domains, sizes
- Emulate realistic changes
  - Remove one constraint → Compile without  
→ Incrementally add removed
  - Apply typical edit patterns e.g.:
    - Add/remove feature
    - Change feature group
    - Move feature

Domain	Models	Features	Clauses	Cardinality
automotive	2	2,513 – 18,616	10,311 – 350,149	$5.3^{210} - 1.7^{1534}$
business	1	1,920	61,362	$2.0^{397}$
cloud	1	106	156	$5.4^{20}$
database	1	117	417	$3.2^1$
deep-learning	2	3,296 – 6,867	8,004 – 17,431	$1.3^{185} - 2.8^{630}$
e-commerce	2	287 – 2,238	421 – 2,615	$2.3^{49} - 8.0^{558}$
finance	3	176 – 771	280 – 7,238	$9.7^{13} - 5.3^{31}$
games	1	144	769	$4.2^{18}$
hardware	2	172 – 377	309 – 1,356	$3.3^{21} - 2.3^{27}$
navigation	2	103 – 145	185 – 378	$3.2^{10} - 1.8^{12}$
security	1	158	457	$1.2^{14}$
systems software	14	96 – 1,580	183 – 15,692	$8.3^{11} - 4.1^{409}$
text	1	137	179	$1.3^{23}$

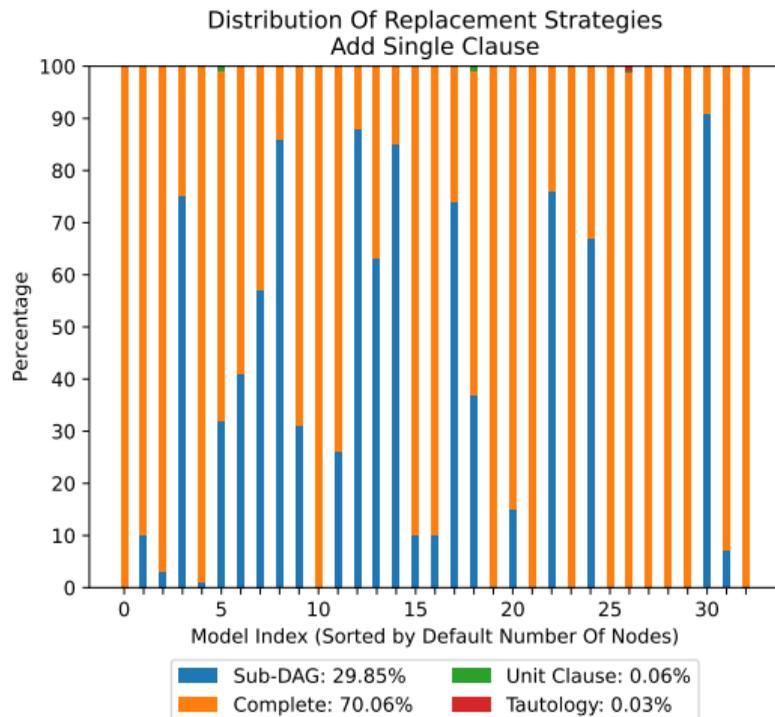
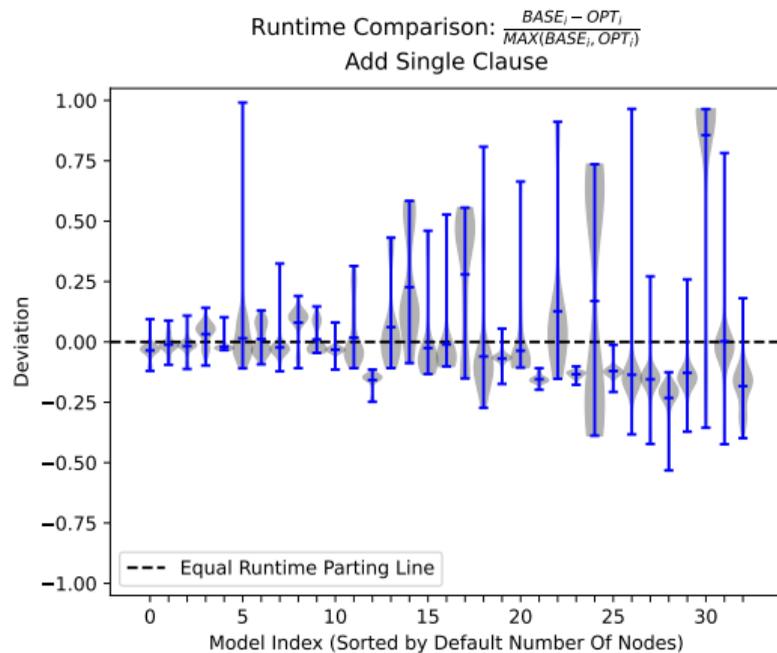
# Evaluation Single Clause



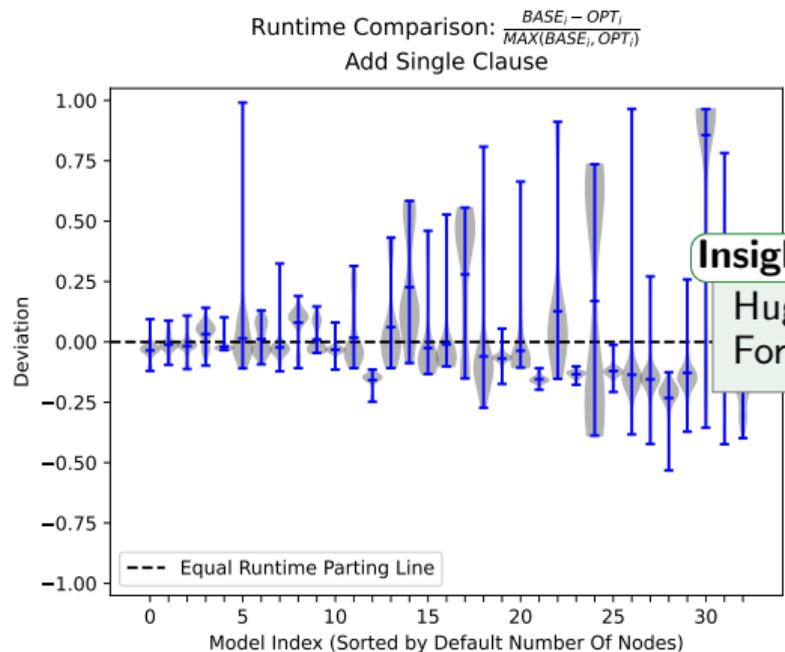
# Evaluation Single Clause



# Evaluation Single Clause

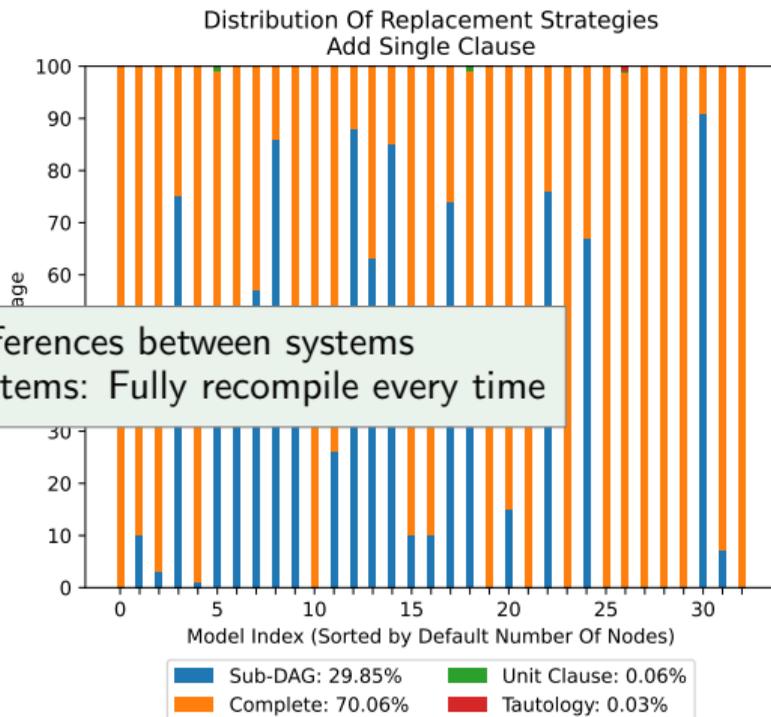


# Evaluation Single Clause



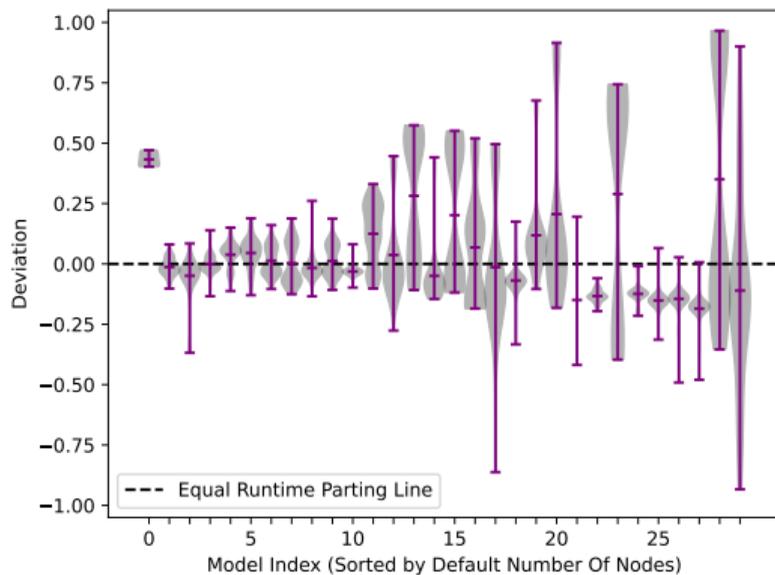
## Insights

Huge differences between systems  
For 6 systems: Fully recompile every time

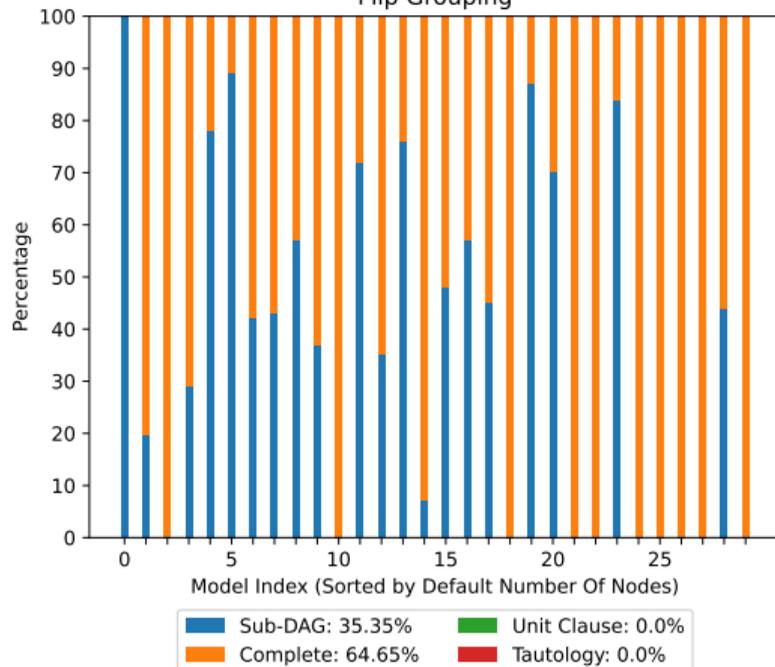


# Evaluation Flip Group

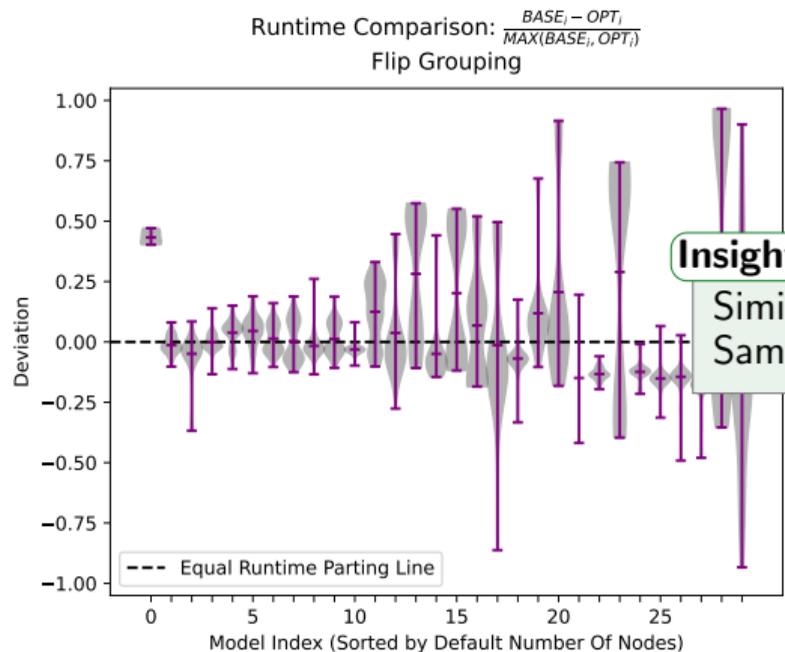
Runtime Comparison:  $\frac{BASE_i - OPT_i}{MAX(BASE_i, OPT_i)}$   
Flip Grouping



Distribution Of Replacement Strategies  
Flip Grouping

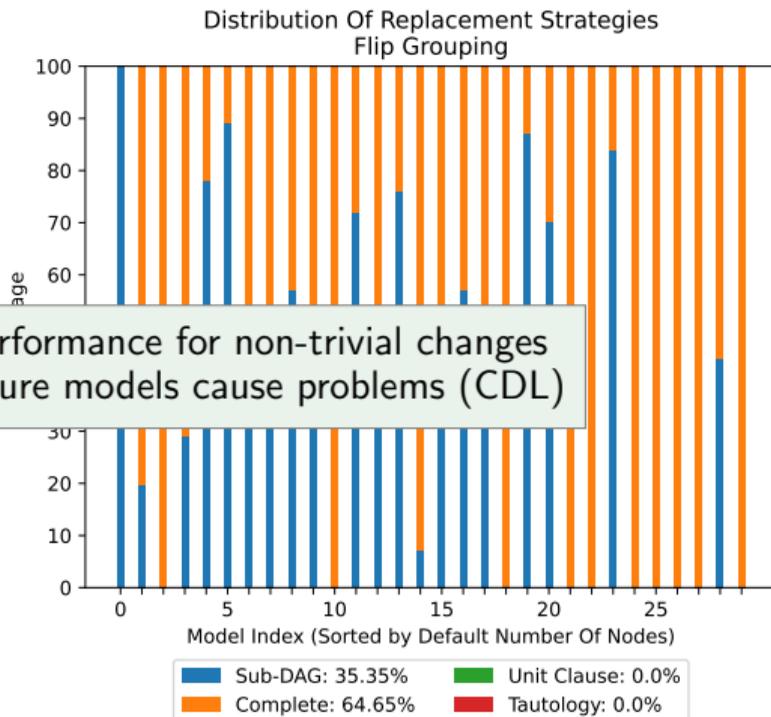


# Evaluation Flip Group



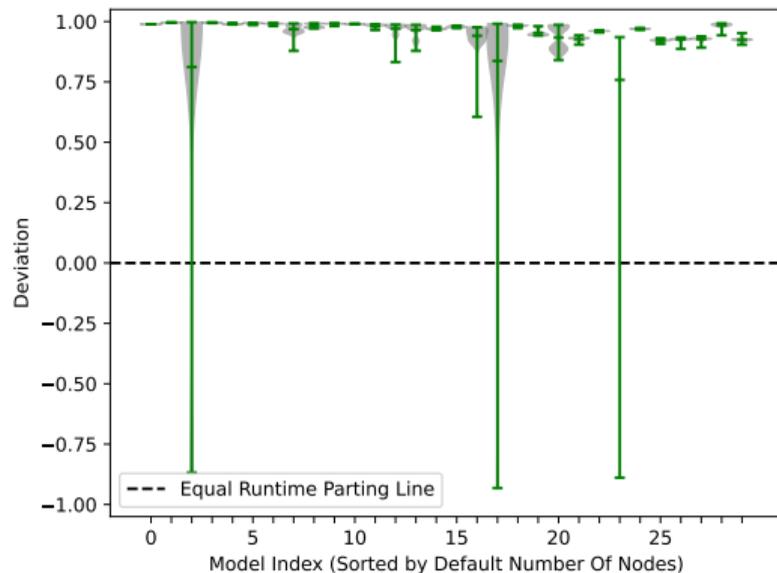
**Insights**

Similar performance for non-trivial changes  
Same feature models cause problems (CDL)

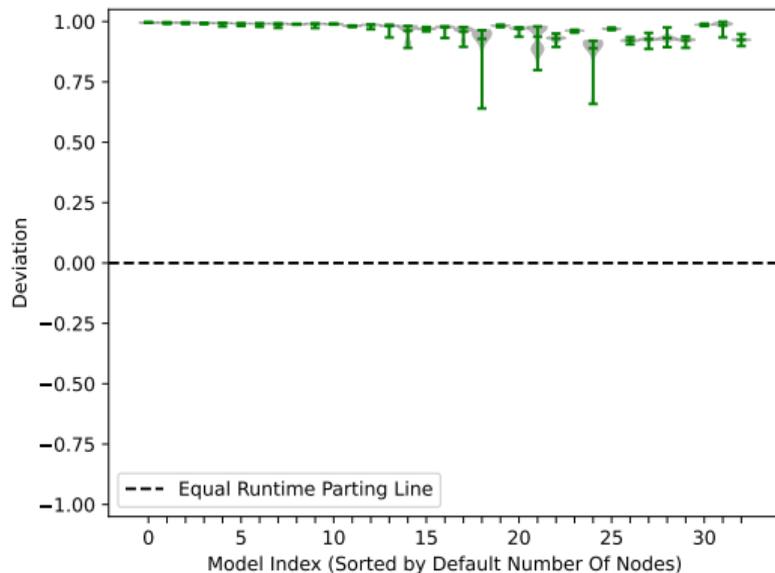


# Evaluation Undo

Runtime Comparison:  $\frac{BASE_j - OPT_j}{MAX(BASE_i, OPT_i)}$   
Undo Grouping

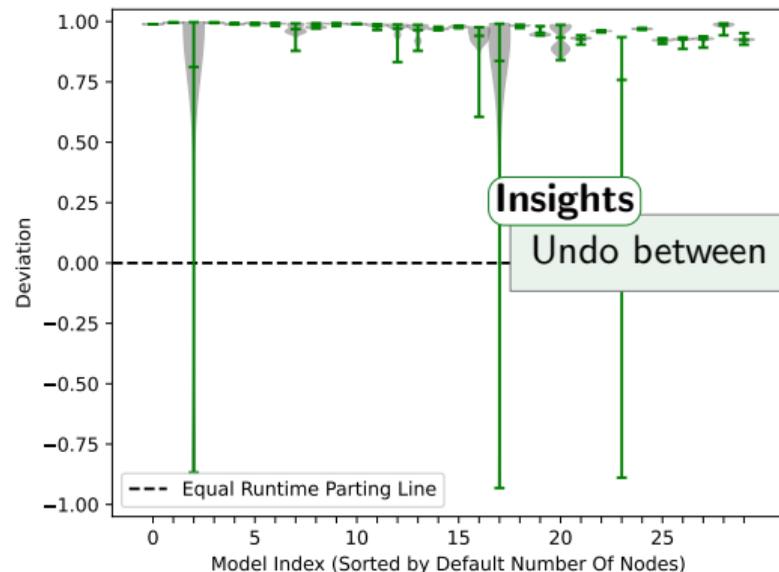


Runtime Comparison:  $\frac{BASE_j - OPT_j}{MAX(BASE_i, OPT_i)}$   
Undo Move Feature



# Evaluation Undo

Runtime Comparison:  $\frac{BASE_i - OPT_i}{MAX(BASE_i, OPT_i)}$   
Undo Grouping



Runtime Comparison:  $\frac{BASE_i - OPT_i}{MAX(BASE_i, OPT_i)}$   
Undo Move Feature



Undo between 23 and 58 times faster than recompiling everytime

# Conclusion & Outlook

## Insights

Substantial runtime improvements for some feature models  
Undo by caching performs very well

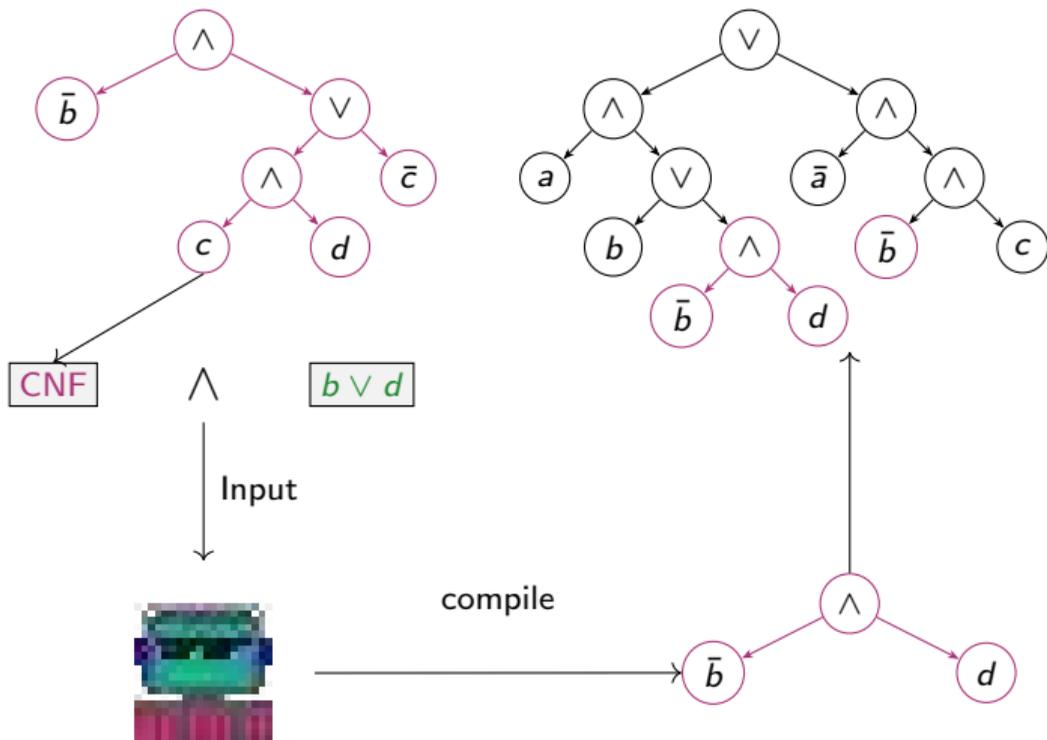
## Limitations

Performance highly depends on input model  
Some models always need full recompilation

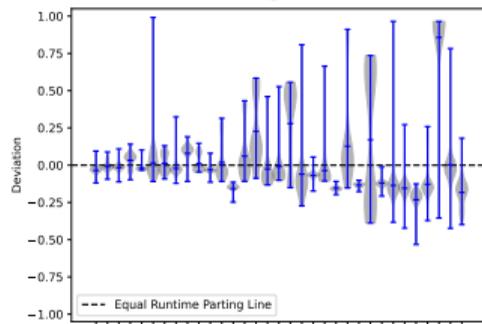
## Future Work

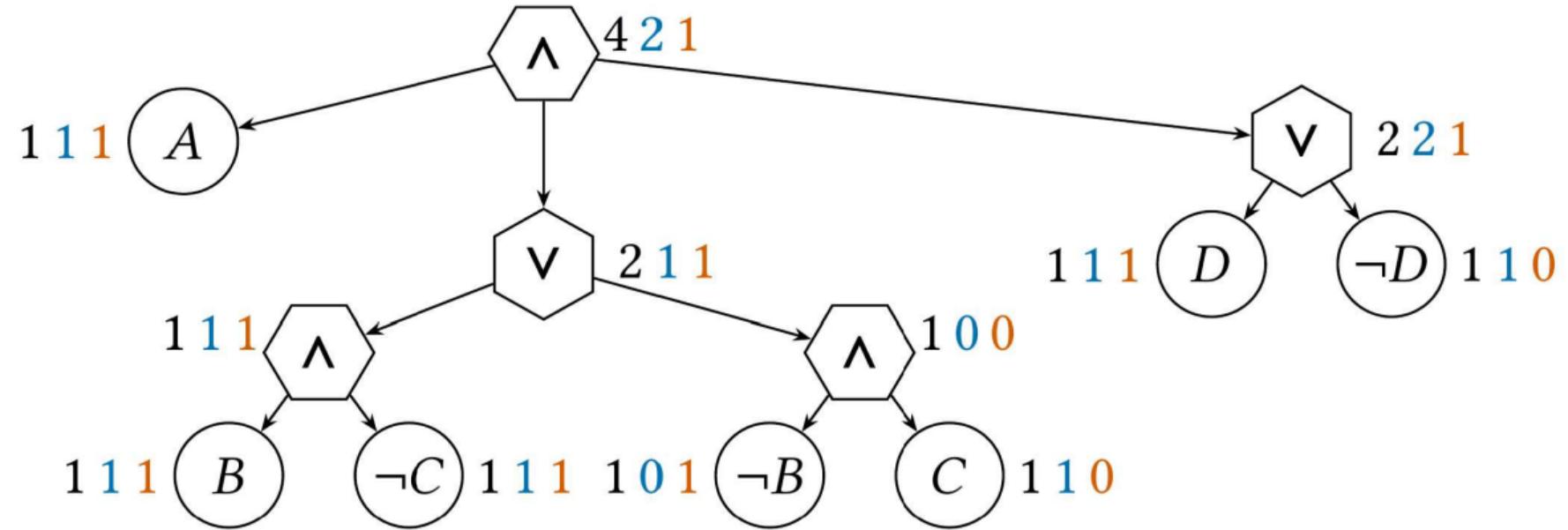
How to improve for unfavourable d-DNNFs?  
How to prepare the initial d-DNNF to be more favourable?  
How to recognize hopeless d-DNNFs?

# Incorporating Feature-Model Edits with Incremental d-DNNF Compilation



Runtime Comparison:  $\frac{BASE}{OPT}$   
Add Single Clause





## Incorporating Feature-Model Edits with Incremental d-DNNF Compilation

FOSD'25 | Chico Sundermann, Heiko Raab, Thomas Thüm | 28.03.2025

# Incorporating Feature-Model Edits with Incremental d-DNNF Compilation

## 1. Motivation

Product Lines  
Feature Dependencies  
Plenty Analyses

Knowledge  
Compilation

## 2. Approach

## 3. Evaluation

Experiment Design

Single Clause

Flip Group

Undo

## 4. Conclusion & Outlook

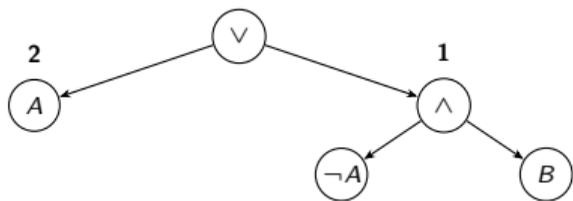
## 5. Future Work

What's left to do?  
Content Overview

# How does this even work? d-DNNFs (Darwiche 2000)

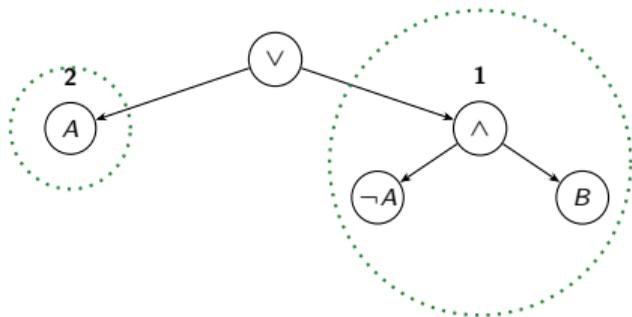
**d**eterministic  
**d**ecomposable  
**n**egation  
**n**ormal  
**f**orm

## How does this even work? d-DNNFs (Darwiche 2000)



**d**eterministic  
**d**ecomposable  
**n**egation  
**n**ormal  
**f**orm

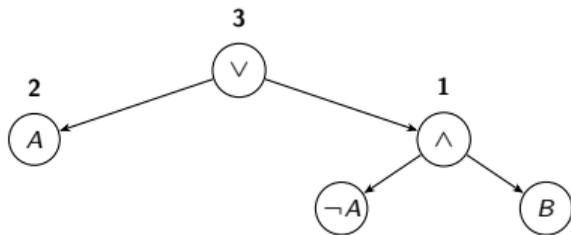
## How does this even work? d-DNNFs (Darwiche 2000)



No shared solutions

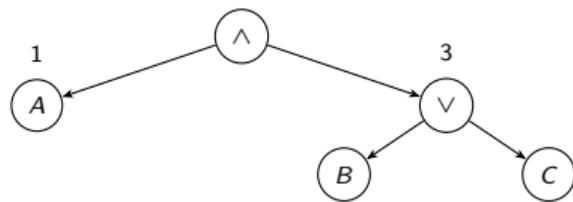
**d**eterministic  
**d**ecomposable  
**n**egation  
**n**ormal  
**f**orm

## How does this even work? d-DNNFs (Darwiche 2000)



**deterministic** (Sum for  $\vee$ )  
**decomposable**  
**negation**  
**normal**  
**form**

## How does this even work? d-DNNFs (Darwiche 2000)



**d**eterministic (Sum for  $\vee$ )

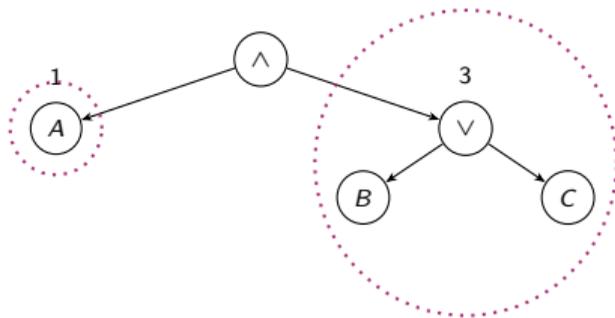
**d**ecomposable

**n**egation

**n**ormal

**f**orm

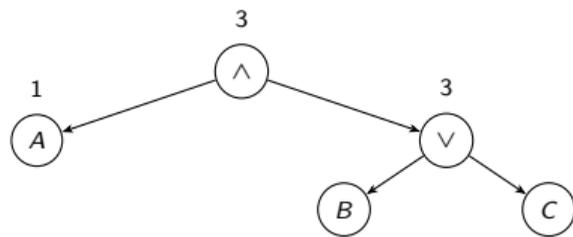
## How does this even work? d-DNNFs (Darwiche 2000)



Independent  
⇒ All pairs valid

deterministic (Sum for  $\vee$ )  
decomposable  
negation  
normal  
form

## How does this even work? d-DNNFs (Darwiche 2000)



**d**eterministic (Sum for  $\vee$ )  
**d**ecomposable (Product for  $\wedge$ )  
**n**egation  
**n**ormal  
**f**orm