Carl von Ossietzky
Universität
Oldenburg

Construction of Verifier Combinations Based on Off-the-Shelf Verifier

Dirk Beyer, Sudeep Kanav, and **Cedric Richter**

04.04.2022

Software Verification

```
int main(){

    unsigned int n = __VERIFIER_nondet_uint();
    unsigned int x=n, y=0;

    while(x>0){
        x--;
        y++;
    }

    __VERIFIER_assert(y==n);
}
```

Verification Task

Software Verification

Program

```
unsigned int n = __VERIFIER_nondet_uint();
unsigned int x=n, y=0;

while(x>0){
    x--;
    y++;
}
```

Software Verification

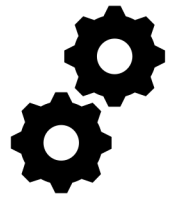
```
int main(){

    unsigned int n = __VERIFIER_nondet_uint();
    unsigned int x=n, y=0;

    while(x>0){
        x--;
        y++;
    }
    __VERIFIER_assert(y==n);
}
```

Specification

Software Verification

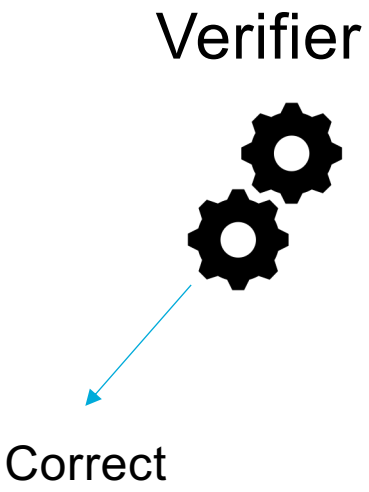
```
int main(){

    unsigned int n = __VERIFIER_nondet_uint();
    unsigned int x=n, y=0;

    while(x>0){
        x--;
        y++;
    }

    __VERIFIER_assert(y==n);
}
```

Verifier



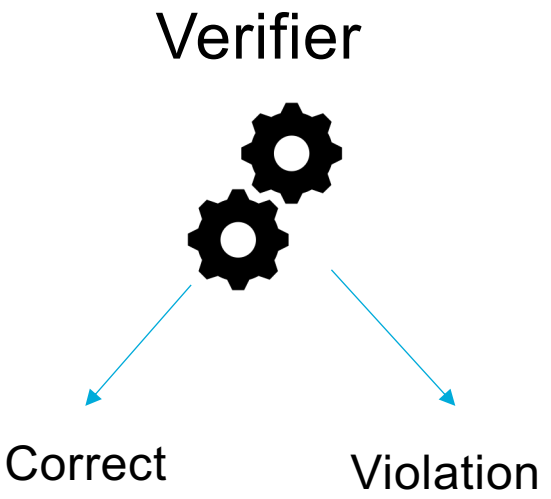
Software Verification

```
int main(){

    unsigned int n = __VERIFIER_nondet_uint();
    unsigned int x=n, y=0;

    while(x>0){
        x--;
        y++;
    }

    __VERIFIER_assert(y==n);
}
```



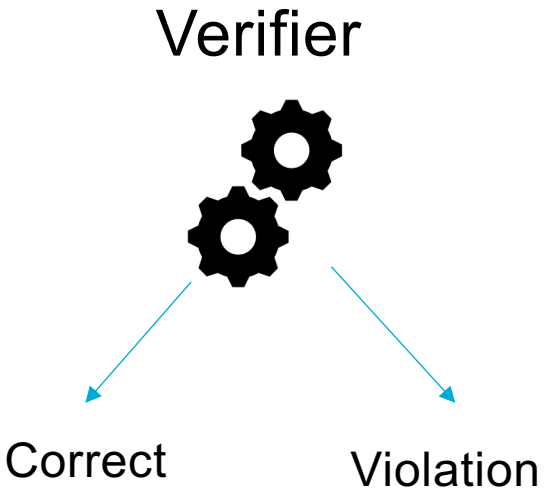
Software Verification

```
int main(){

    unsigned int n = __VERIFIER_nondet_uint();
    unsigned int x=n, y=0;

    while(x>0){
        x--;
        y++;
    }

    __VERIFIER_assert(y==n);
}
```

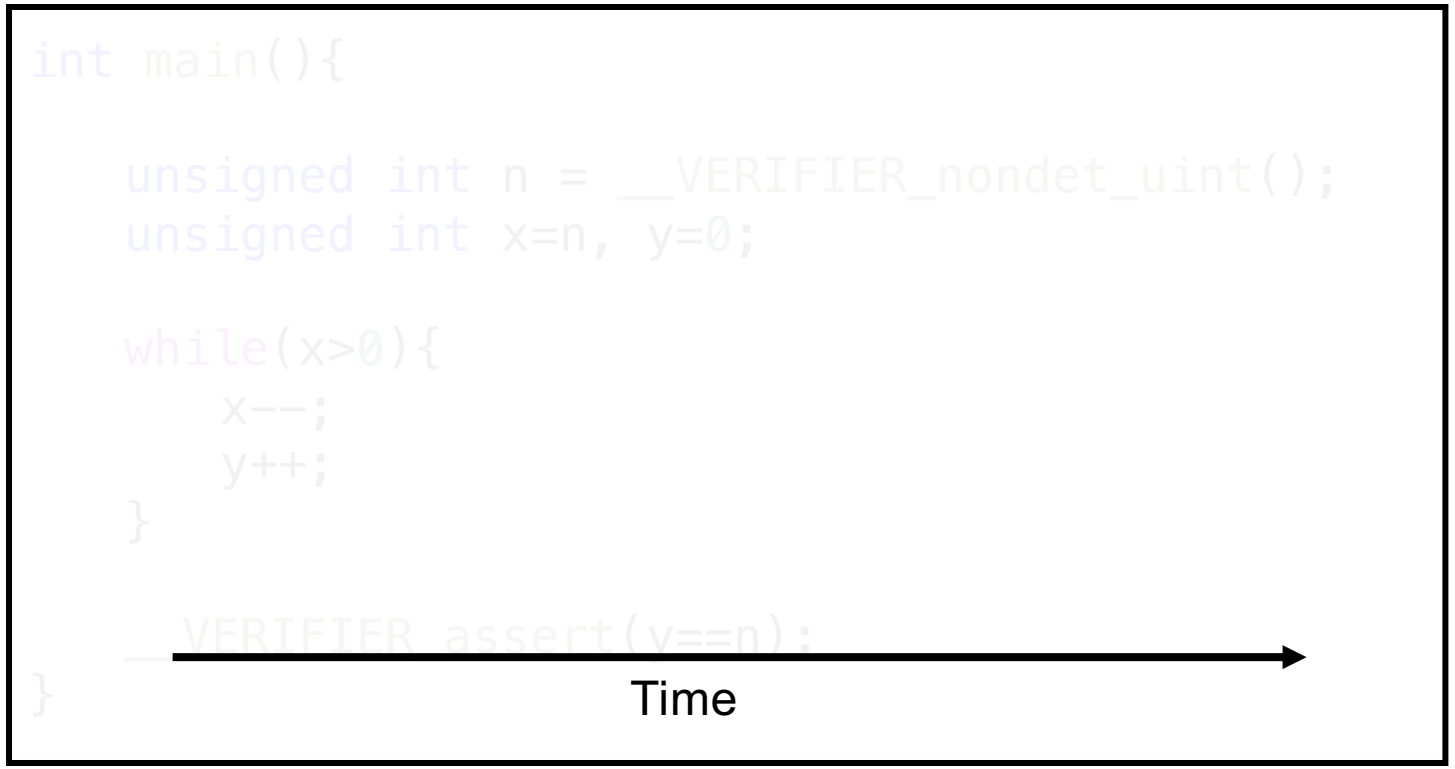


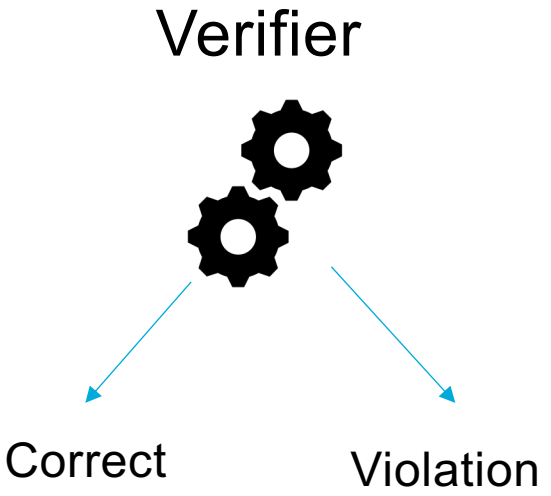
Realistic Setting

```
int main(){

    unsigned int n = __VERIFIER_nondet_uint();
    unsigned int x=n, y=0;

    while(x>0){
        x--;
        y++;
    }

    __VERIFIER_assert(y==n);
}
```

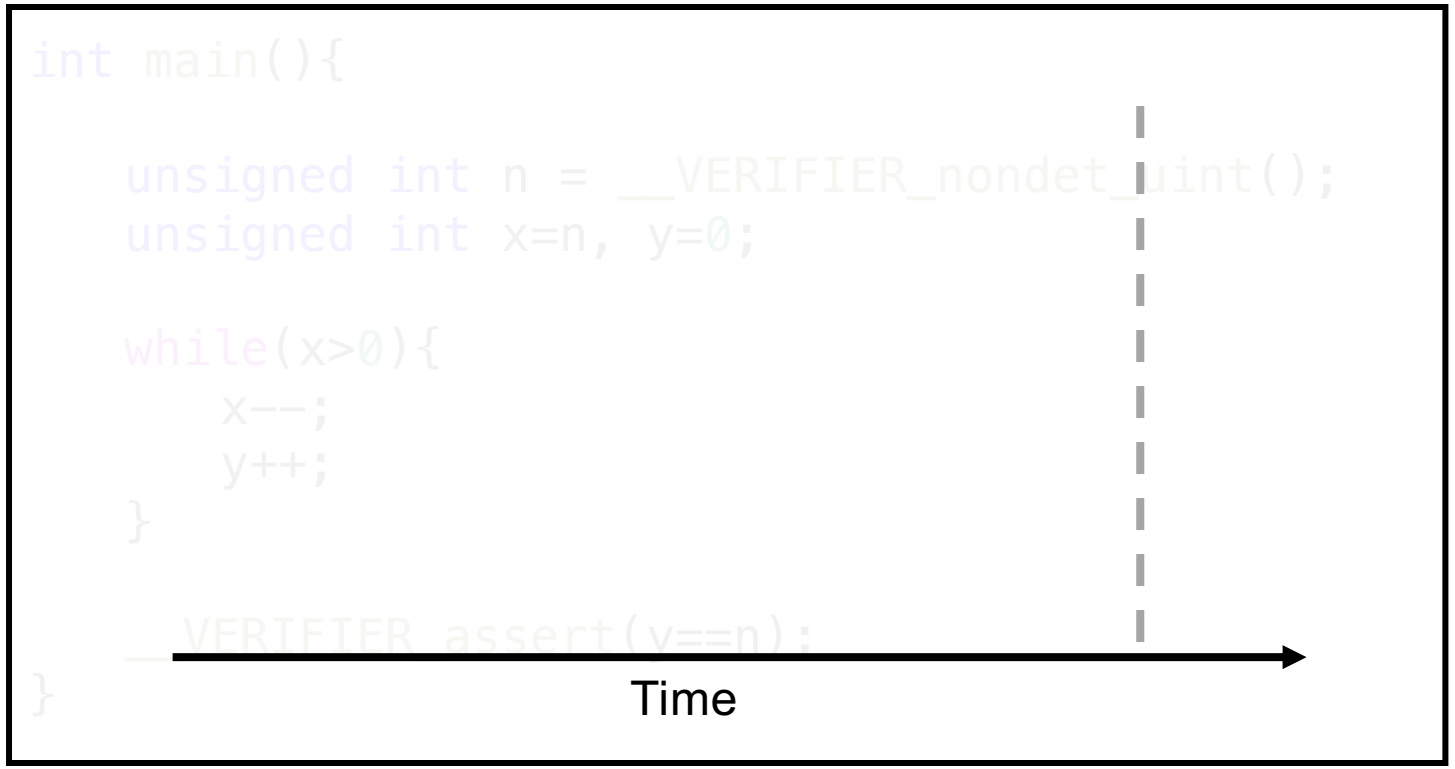


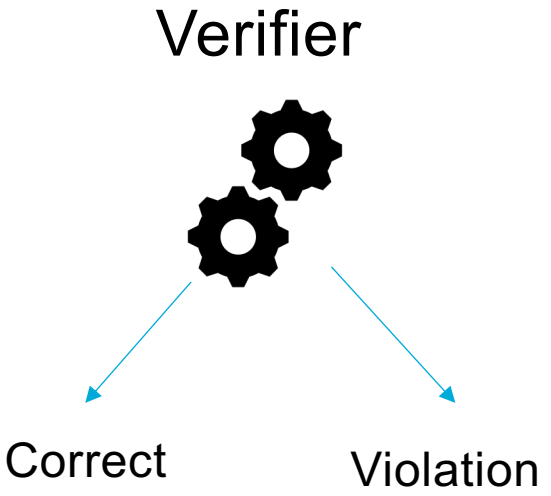
Realistic Setting



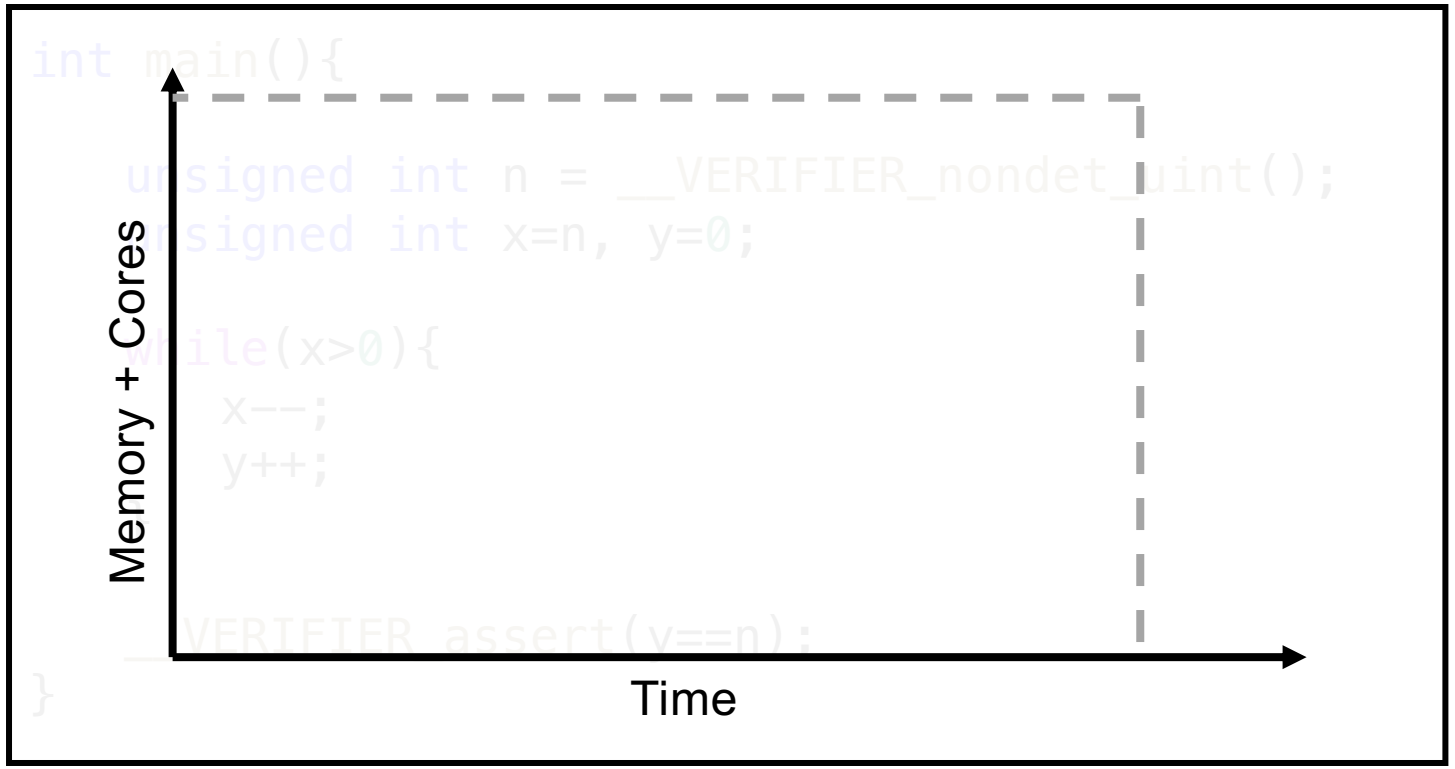


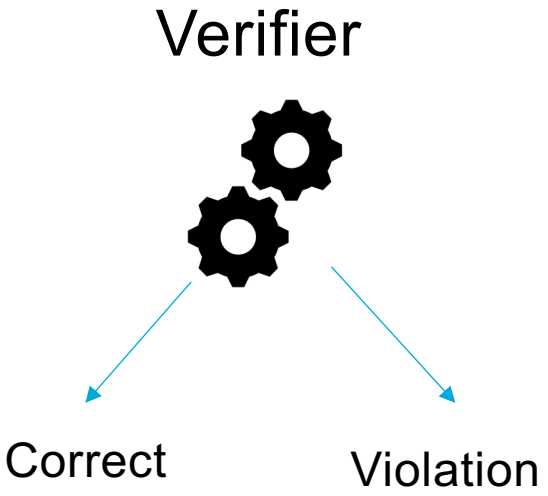
Realistic Setting



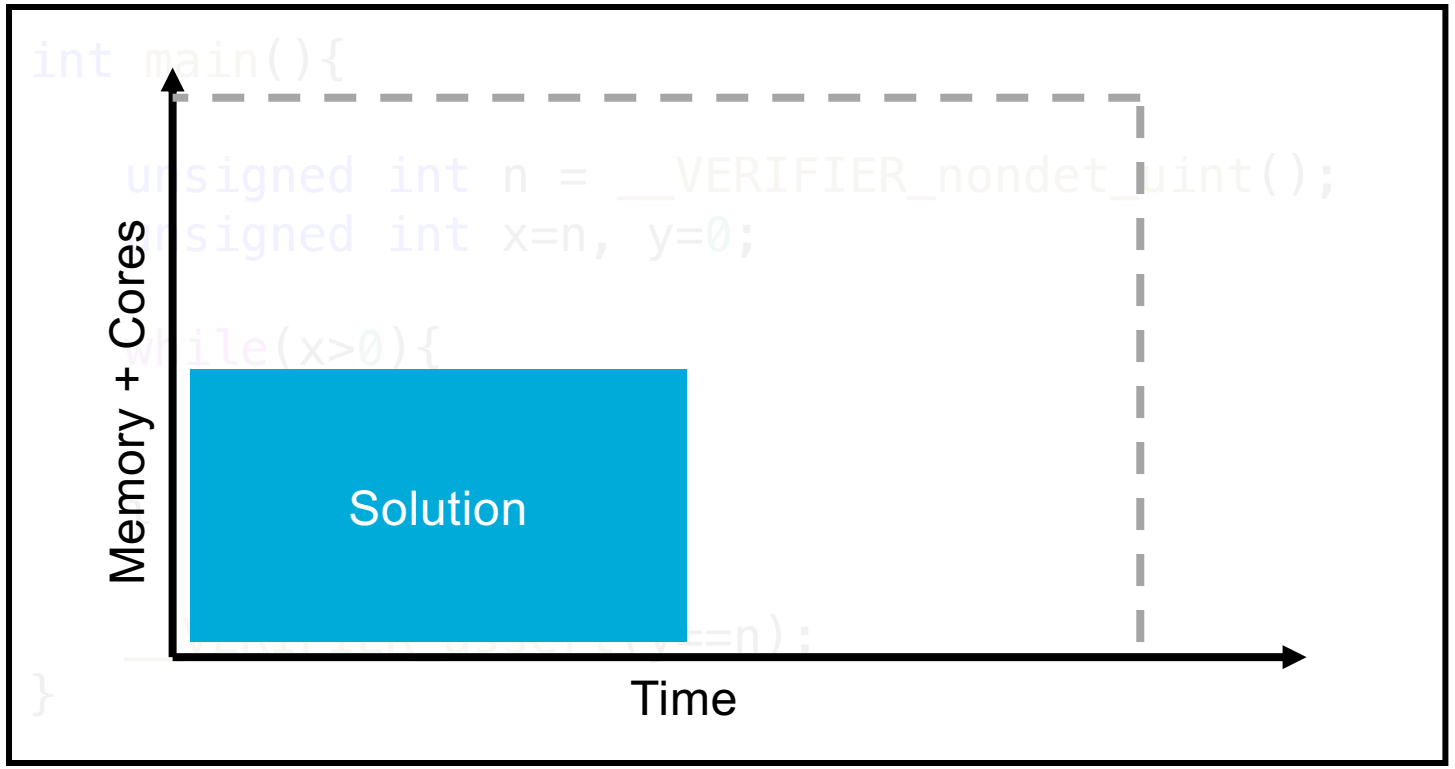


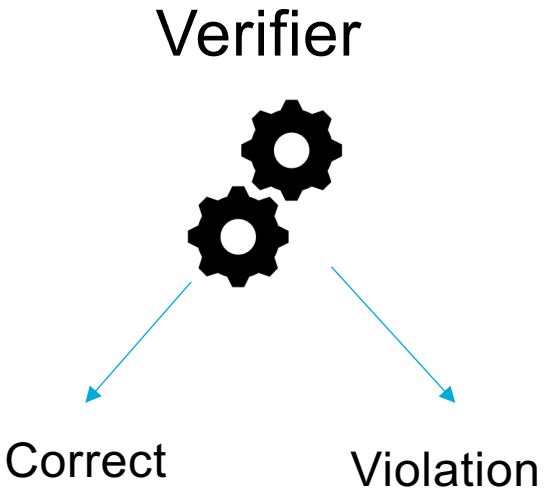
Realistic Setting



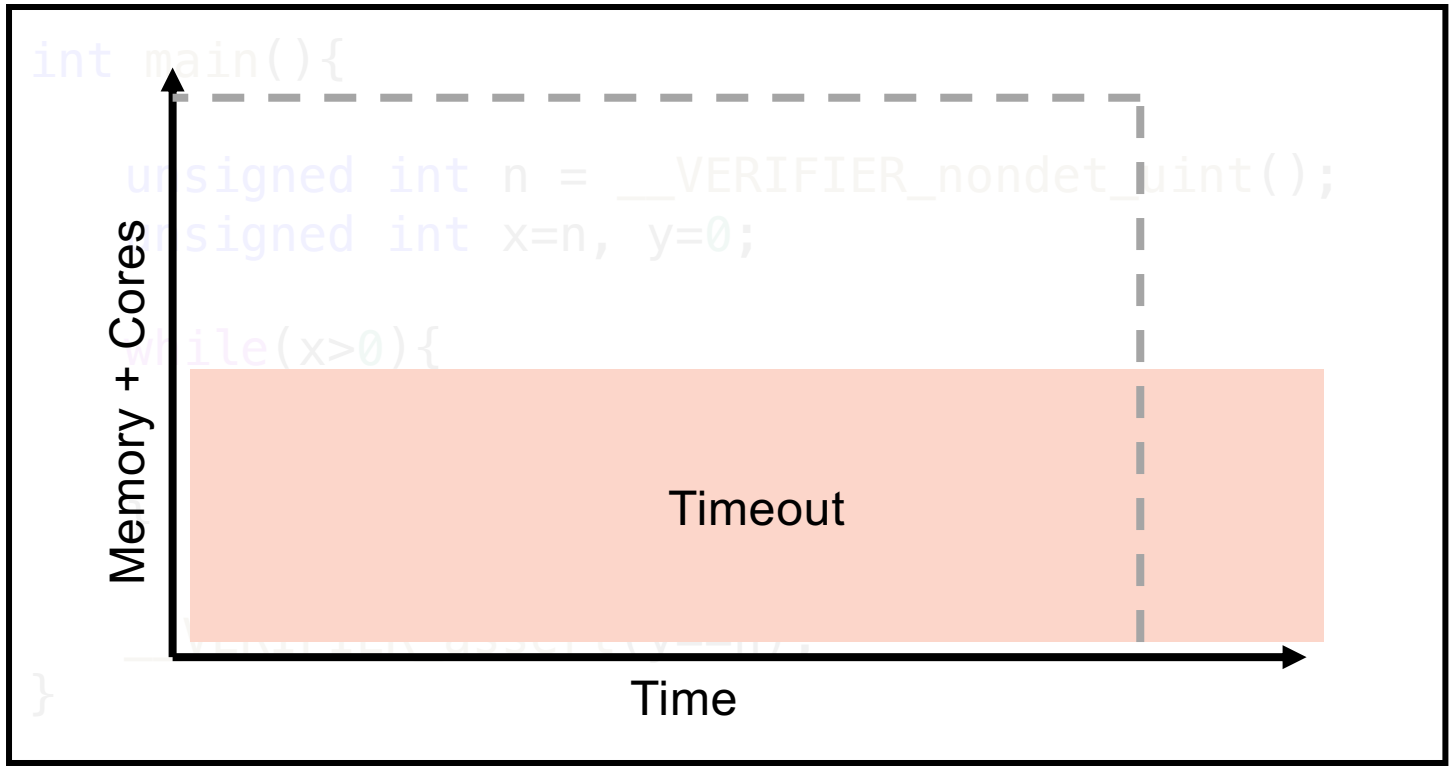


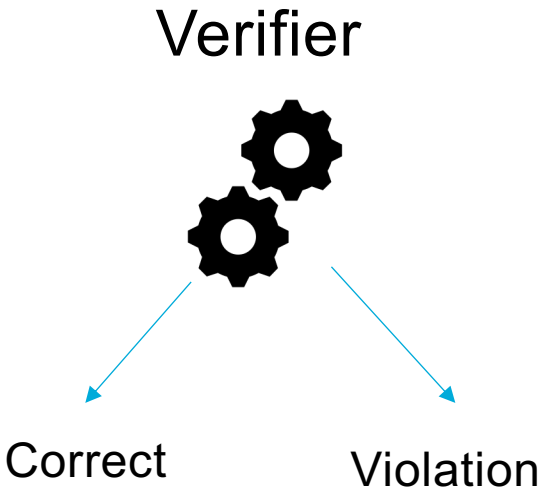
Realistic Setting



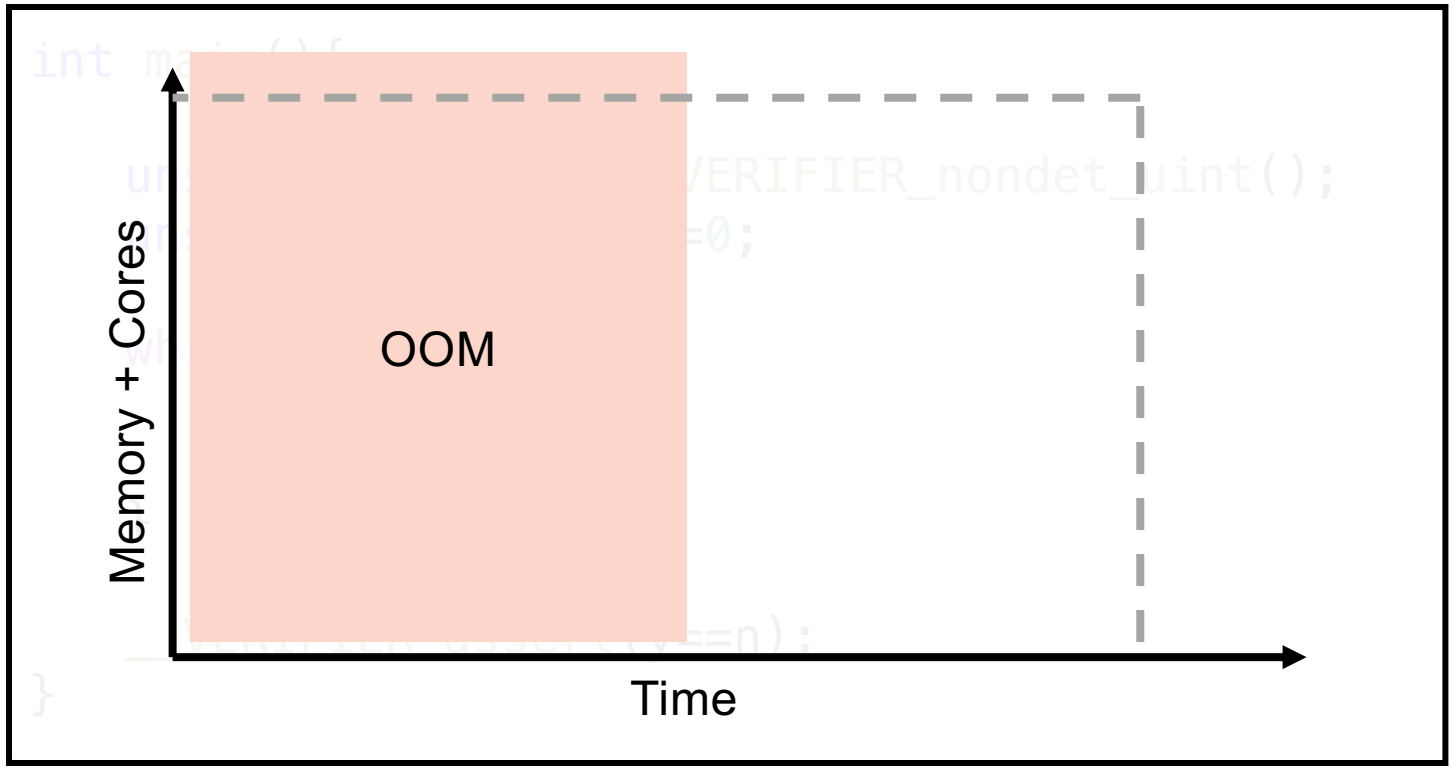


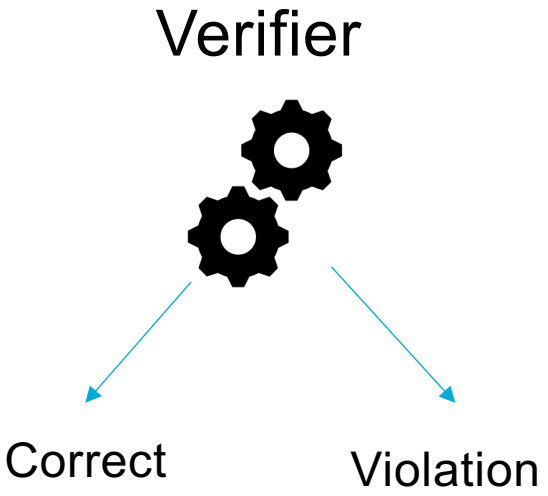
Realistic Setting



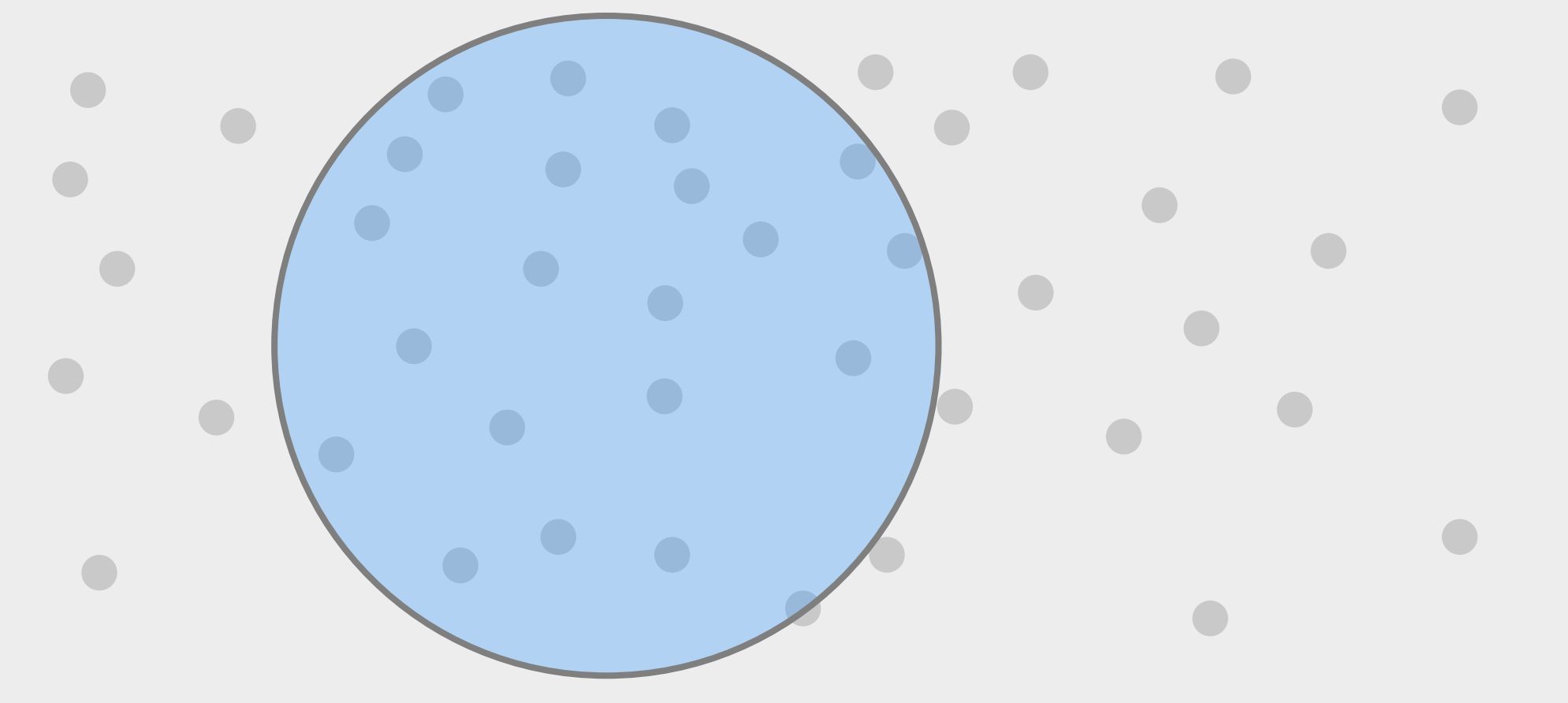


Realistic Setting



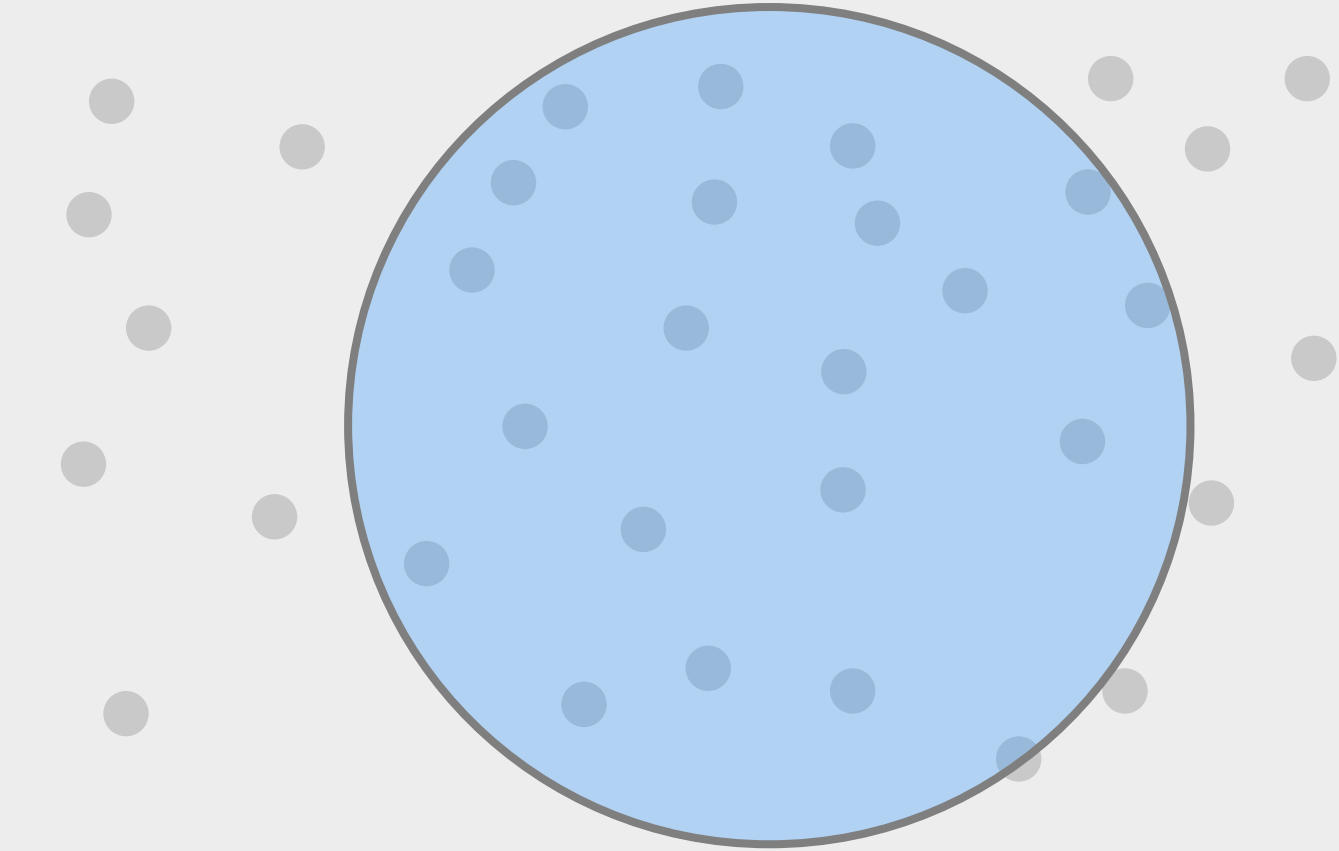


Limits of a **single** verifier

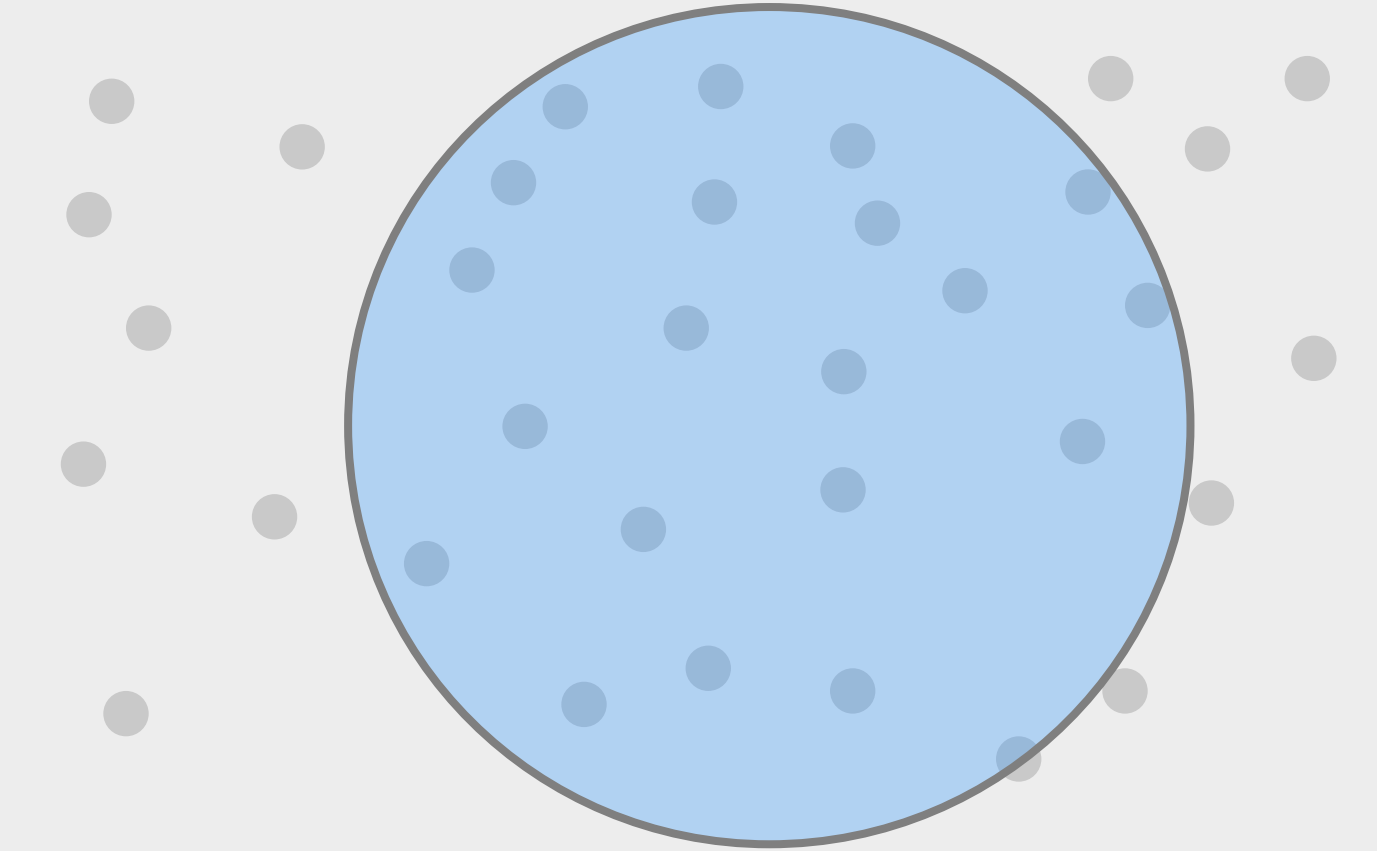


CPAchecker

Limits of a **single** verifier



Limits of a **single** verifier



CPAchecker

UAutomizer

ESBMC

Symbiotic

CBMC

Divine

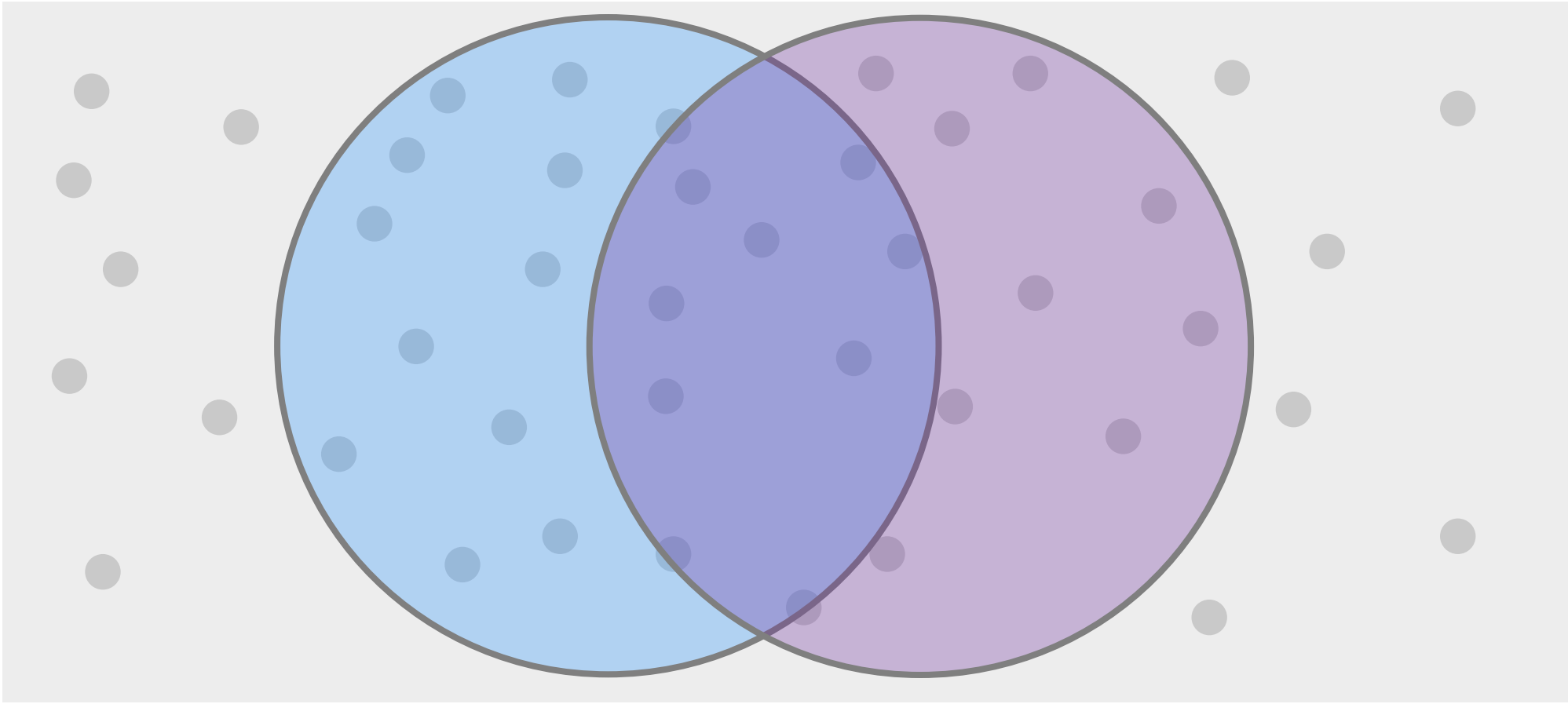
Goblint

UTaipan

⋮

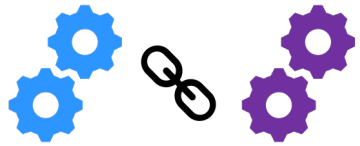
Verification Task

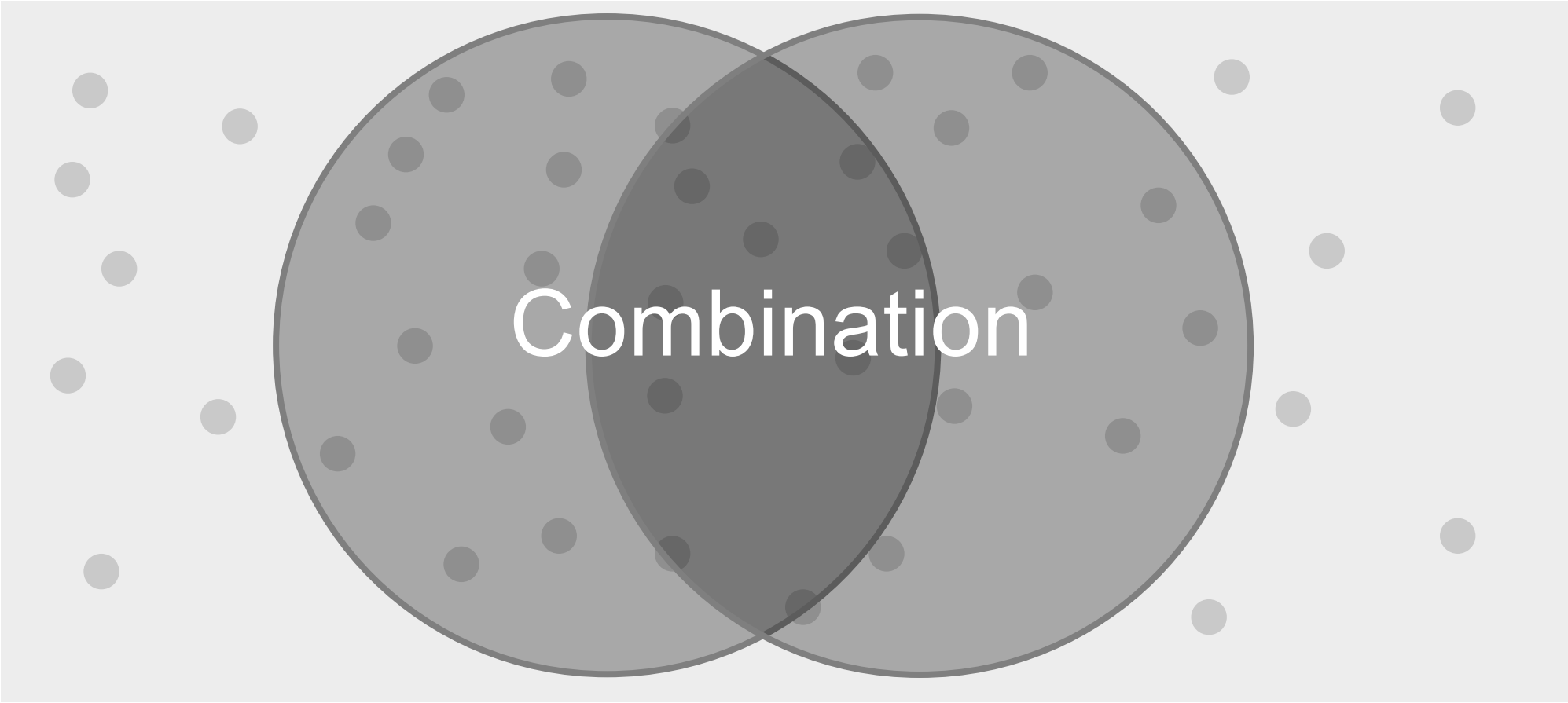
Limits of a **single** verifier



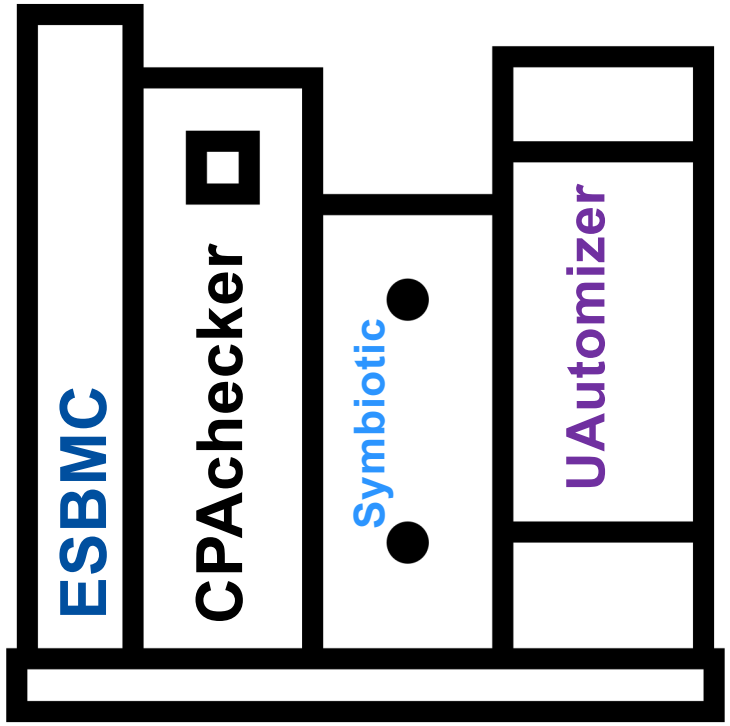
Verification Task

Can we improve a single verifier **by combination**?

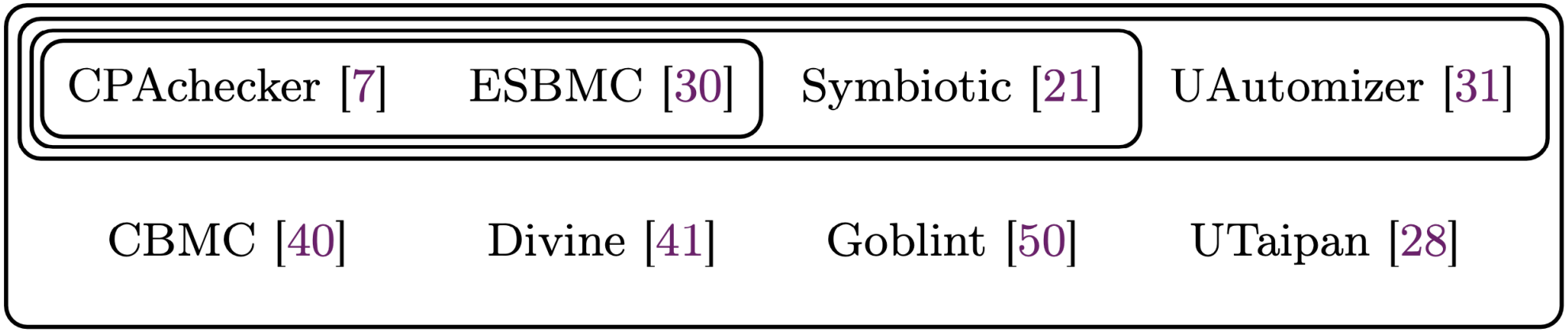




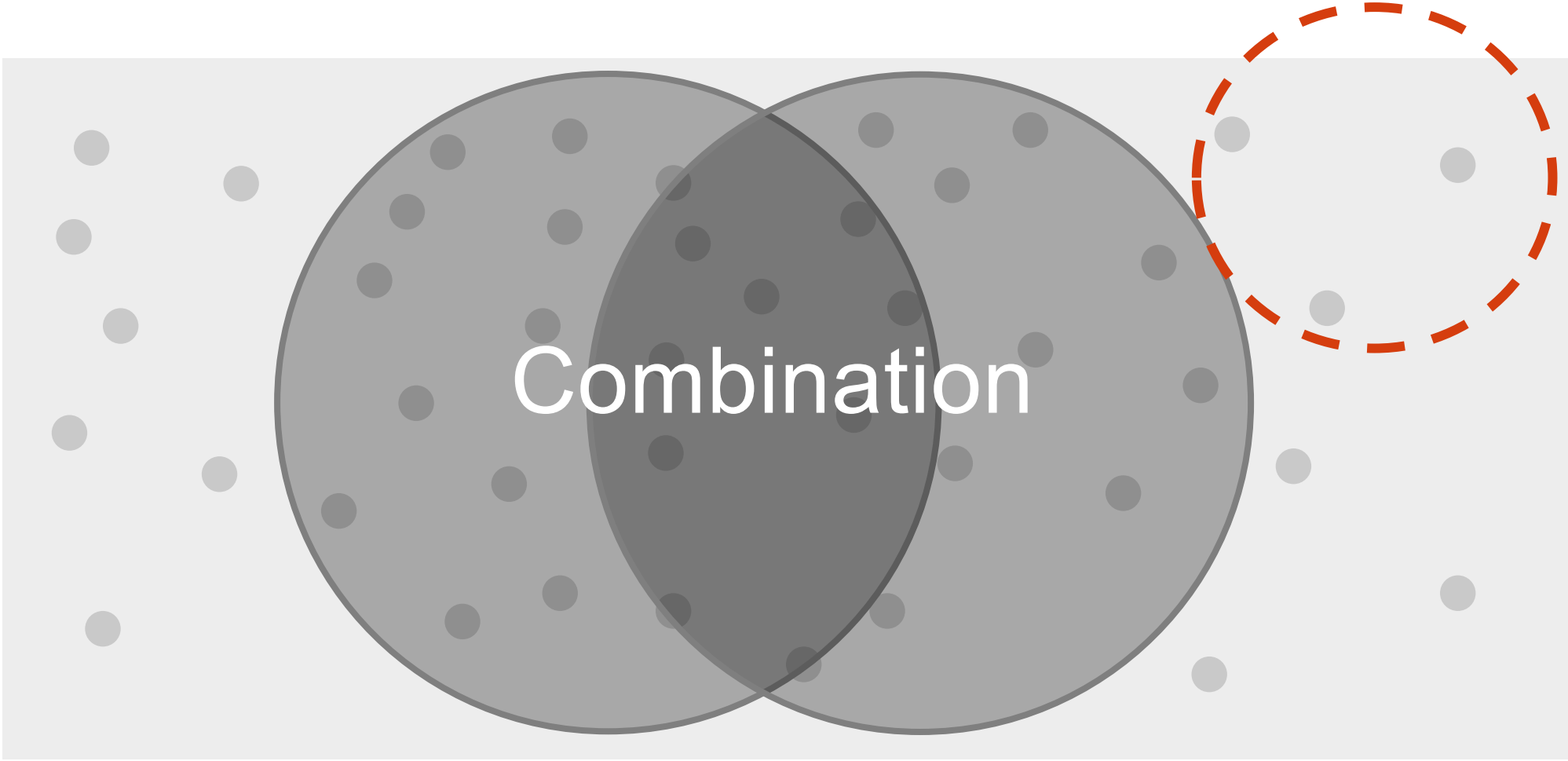
Off-the-shelf construction of combinations



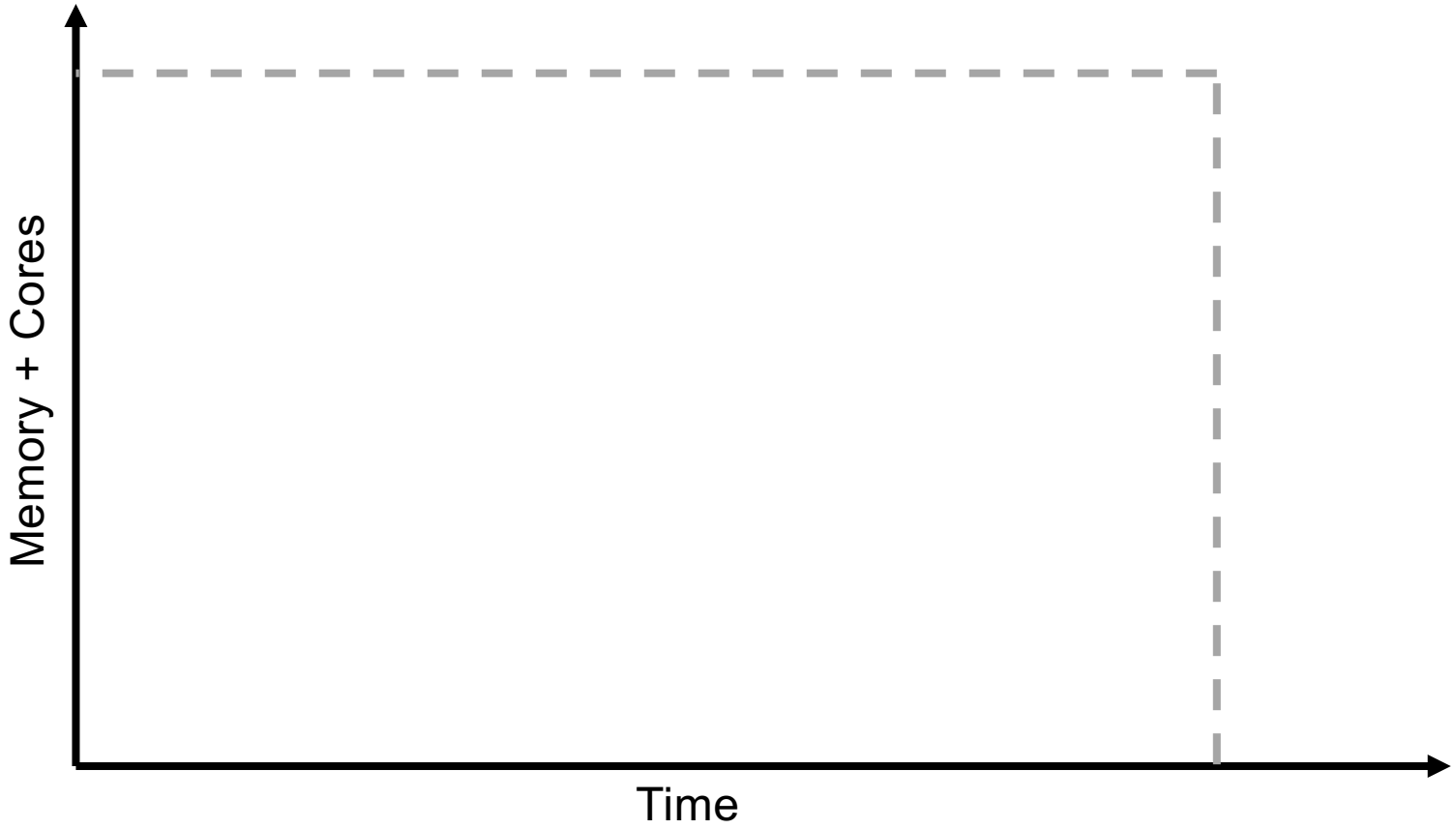
Verifier Selection



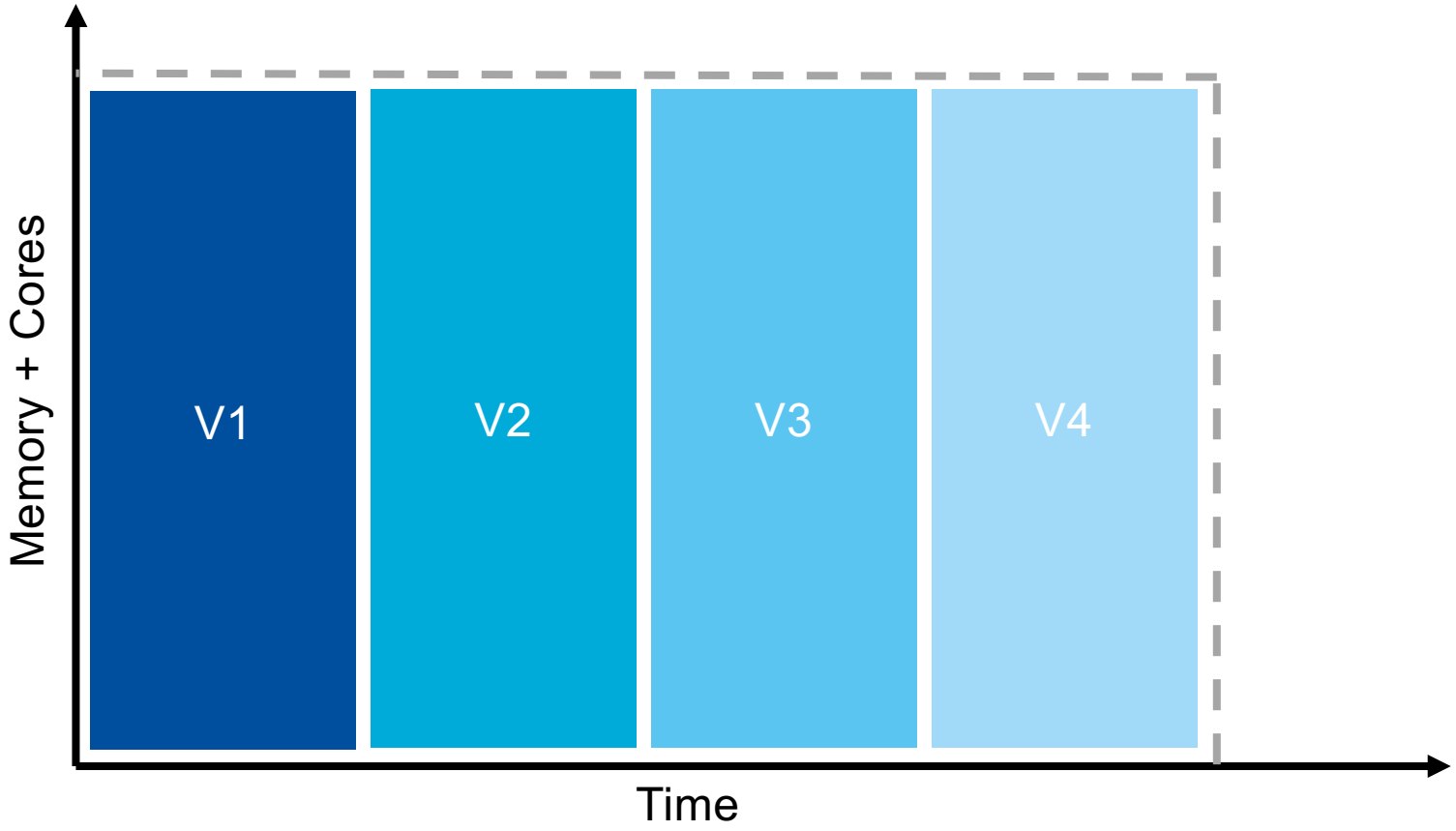
Theoretical **Limit**



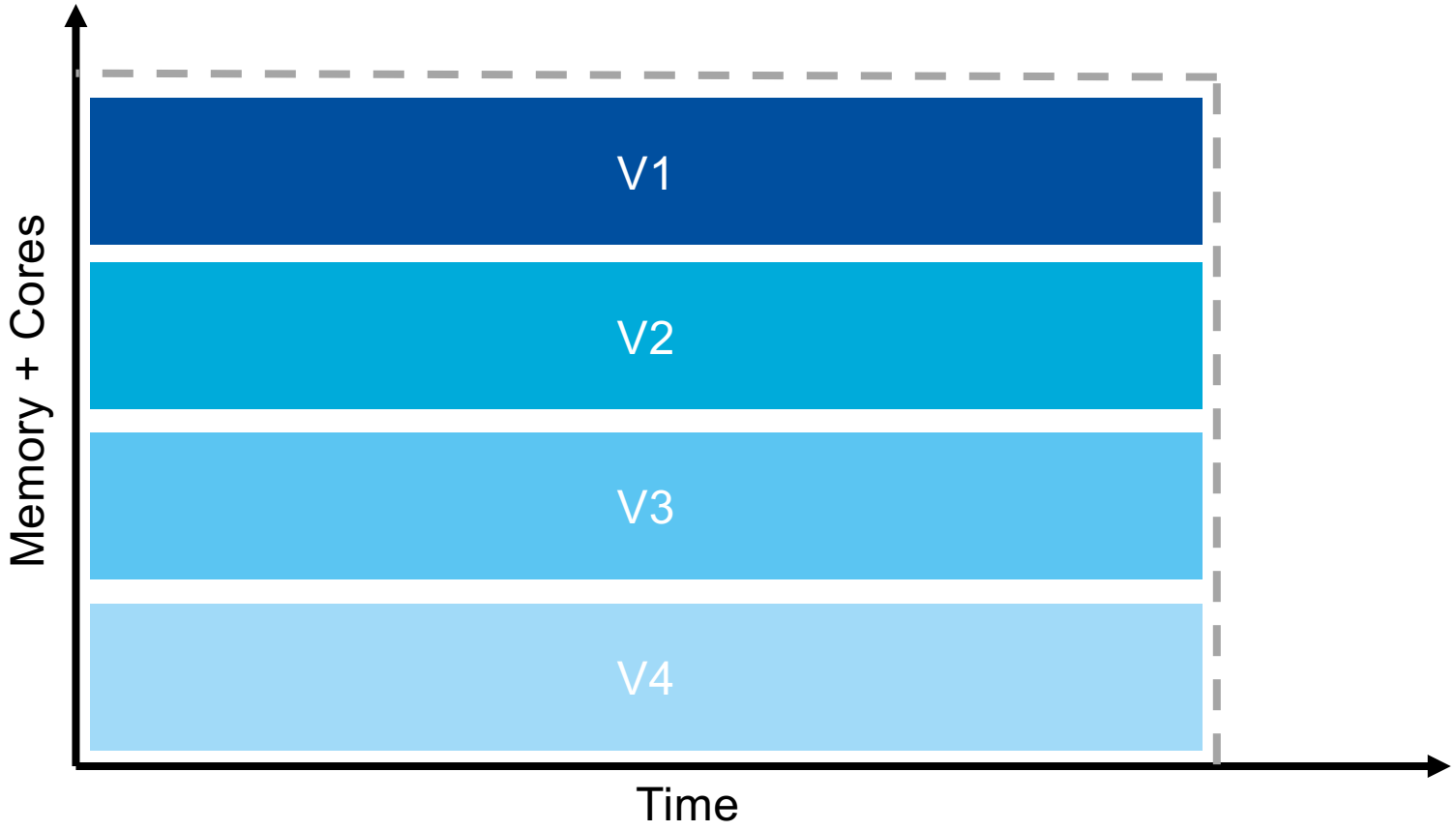
Black-box **combinations**



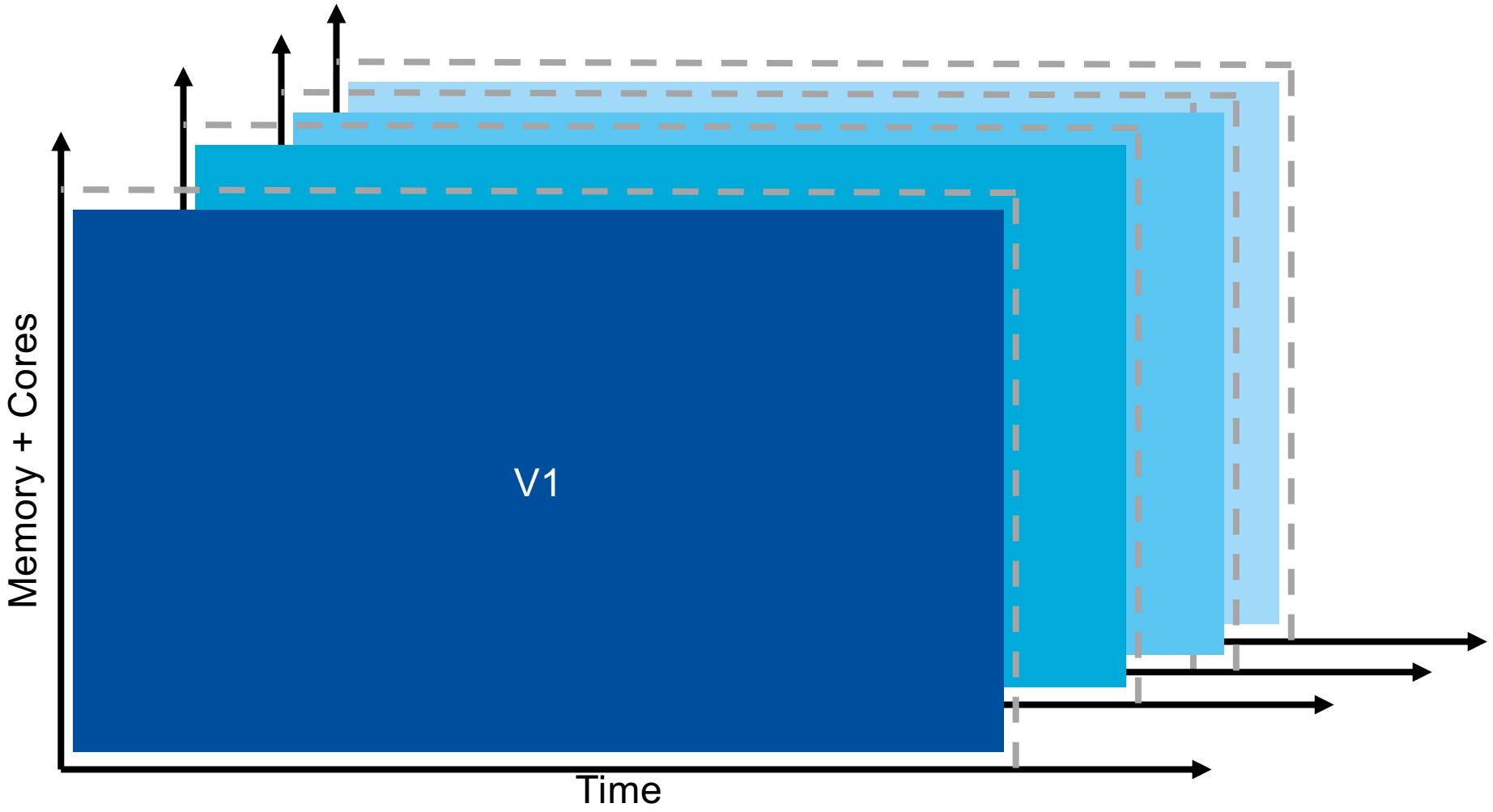
Sequential **portfolios**



Parallel **portfolios**

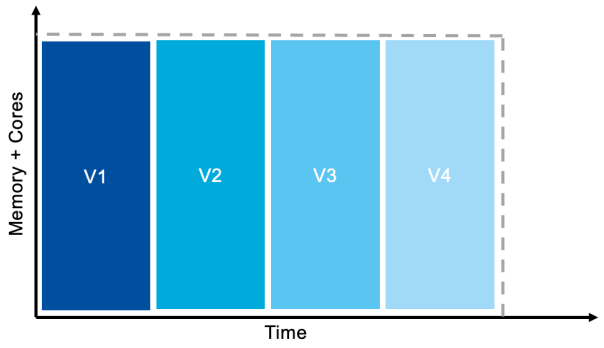


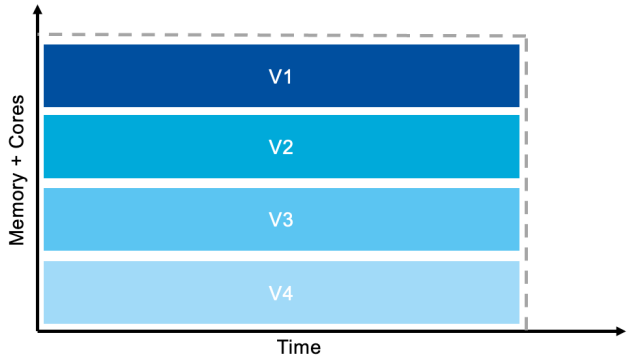
Algorithm **selection**

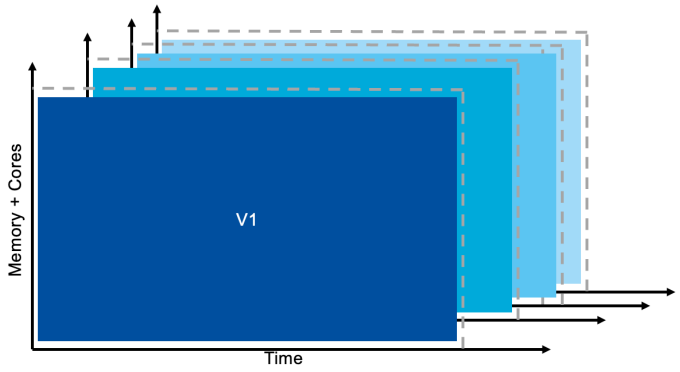


Evaluation

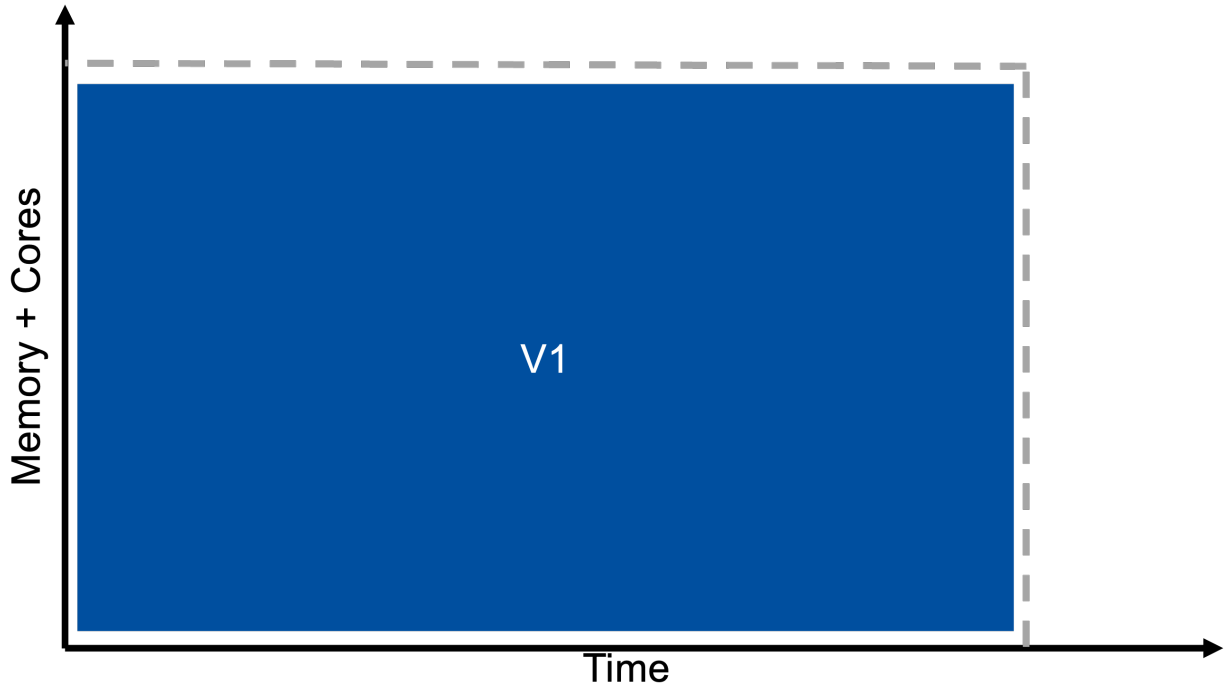
Integrated into **CoVeriTeam**







Can we improve a single verifier **by combination**?



Single Verifier

Single Verifier

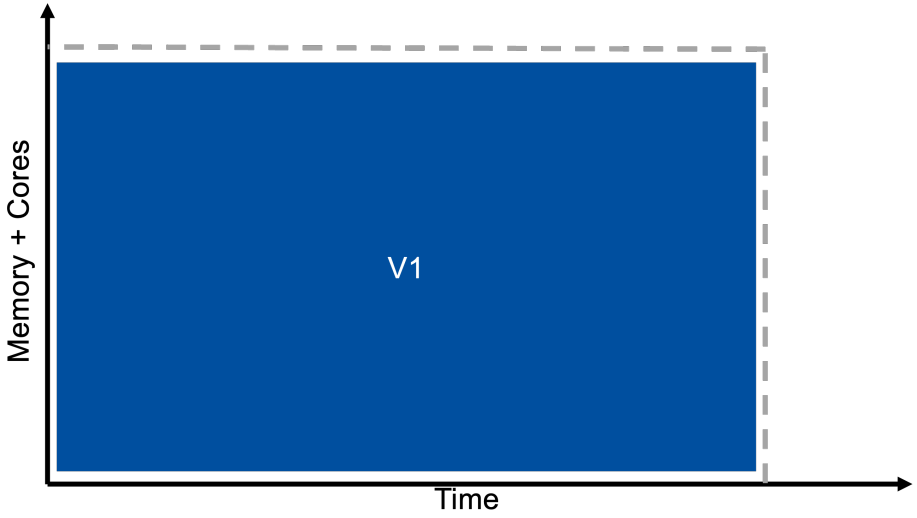
Verifier	CPAchecker	EsBMC	Symbiotic	UAutomizer	CBMC	Divine	Goblint	UTaipan
Score	**9 040**	6 623	4 878	7 146	4 663	3 679	2 770	5 338
Correct results	**5 652**	4 481	3 001	4 358	3 484	2 922	1 385	3 725
Correct proofs	**3 516**	2 958	1 909	2 836	1 499	1 605	1 385	2 365
Correct alarms	2 136	1 523	1 092	1 522	1 985	1 317	0	1 360
Wrong results	8	29	2	2	19	41	**0**	24
Wrong proofs	**0**	22	**0**	1	1	12	**0**	23
Wrong alarms	8	7	2	**1**	18	29	0	**1**

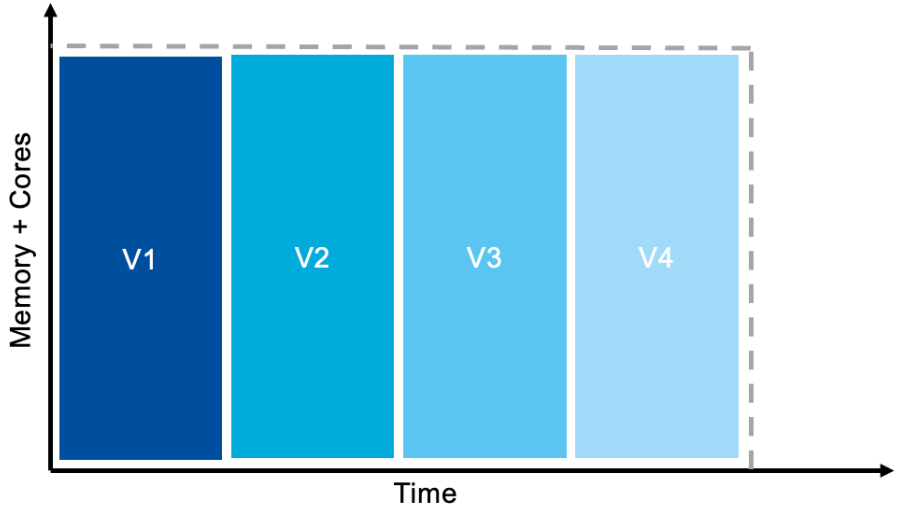
*Verifiers are taken from SVComp 2021

Single Verifier

Verifier	CPAchecker	ESBMC	Symbiotic	UAutomizer	CBMC	Divine	Goblint	UTaipan
Score	**9 040**	6 623	4 878	7 146	4 663	3 679	2 770	5 338
Correct results	**5 652**	4 481	3 001	4 358	3 484	2 922	1 385	3 725
Correct proofs	**3 516**	2 958	1 909	2 836	1 499	1 605	1 385	2 365
Correct alarms	2 136	1 523	1 092	1 522	1 985	1 317	0	1 360
Wrong results	8	29	2	2	19	41	**0**	24
Wrong proofs	**0**	22	**0**	1	1	12	**0**	23
Wrong alarms	8	7	2	**1**	18	29	0	**1**

Can we improve a single verifier **by sequential portfolios**?





Sequential **Portfolio**

Verifier	CPAchecker	Sequential Portfolio of 2	Sequential Portfolio of 3	Sequential Portfolio of 4	Sequential Portfolio of 8
Score	9 040	9 198	9 519	**9 522**	8 349
Correct results	5 652	6 058	6 239	**6 275**	6 084
Correct proofs	3 516	3 780	**3 920**	3 903	3 721
Correct alarms	2 136	2 278	2 319	**2 372**	2 363
Wrong results	**8**	26	26	27	61
Wrong proofs	**0**	14	14	14	30
Wrong alarms	**8**	12	12	13	31

A portfolio of **two** already improves the performance

Verifier	CPAchecker	Sequential Portfolio of 2	3	4	8
Score	9 040	9 198	9 519	**9 522**	8 349
Correct results	5 652	6 058	6 239	**6 275**	6 084
Correct proofs	3 516	3 780	**3 920**	3 903	3 721
Correct alarms	2 136	2 278	2 319	**2 372**	2 363
Wrong results	**8**	26	26	27	61
Wrong proofs	**0**	14	14	14	30
Wrong alarms	**8**	12	12	13	31

CPAchecker [7] ESBMC [30]

More verifiers lead to further improvements

Verifier	CPAchecker	Sequential Portfolio of 2	Sequential Portfolio of 3	Sequential Portfolio of 4	Sequential Portfolio of 8
Score	9 040	9 198	9 519	**9 522**	8 349
Correct results	5 652	6 058	6 239	**6 275**	6 084
Correct proofs	3 516	3 780	**3 920**	3 903	3 721
Correct alarms	2 136	2 278	2 319	**2 372**	2 363
Wrong results	**8**	26	26	27	61
Wrong proofs	**0**	14	14	14	30
Wrong alarms	**8**	12	12	13	31

CPAchecker [7] ESBMC [30] Symbiotic [21] UAutomizer [31]

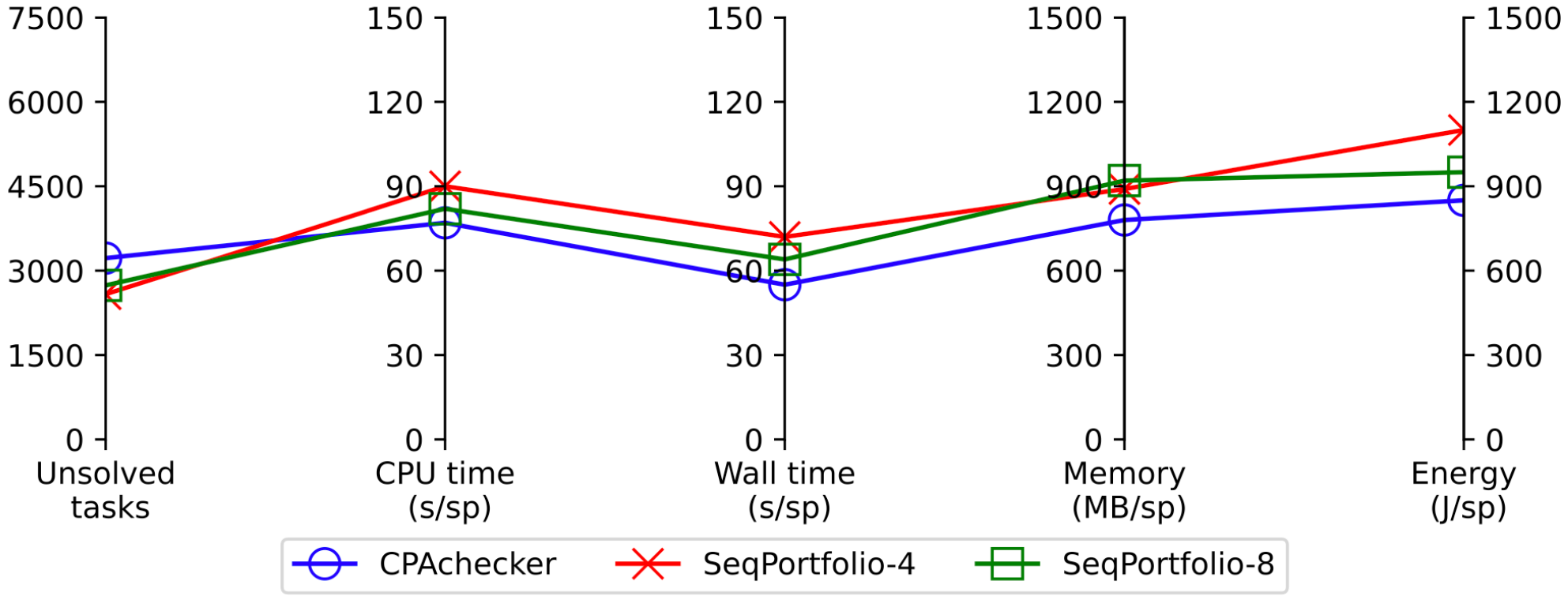
Limited by time

Verifier	CPAchecker	Sequential Portfolio of 2	Sequential Portfolio of 3	Sequential Portfolio of 4	Sequential Portfolio of 8
Score	9 040	9 198	9 519	**9 522**	8 349
Correct results	5 652	6 058	6 239	**6 275**	6 084
Correct proofs	3 516	3 780	**3 920**	3 903	3 721
Correct alarms	2 136	2 278	2 319	**2 372**	2 363
Wrong results	**8**	26	26	27	61
Wrong proofs	**0**	14	14	14	30
Wrong alarms	**8**	12	12	13	31

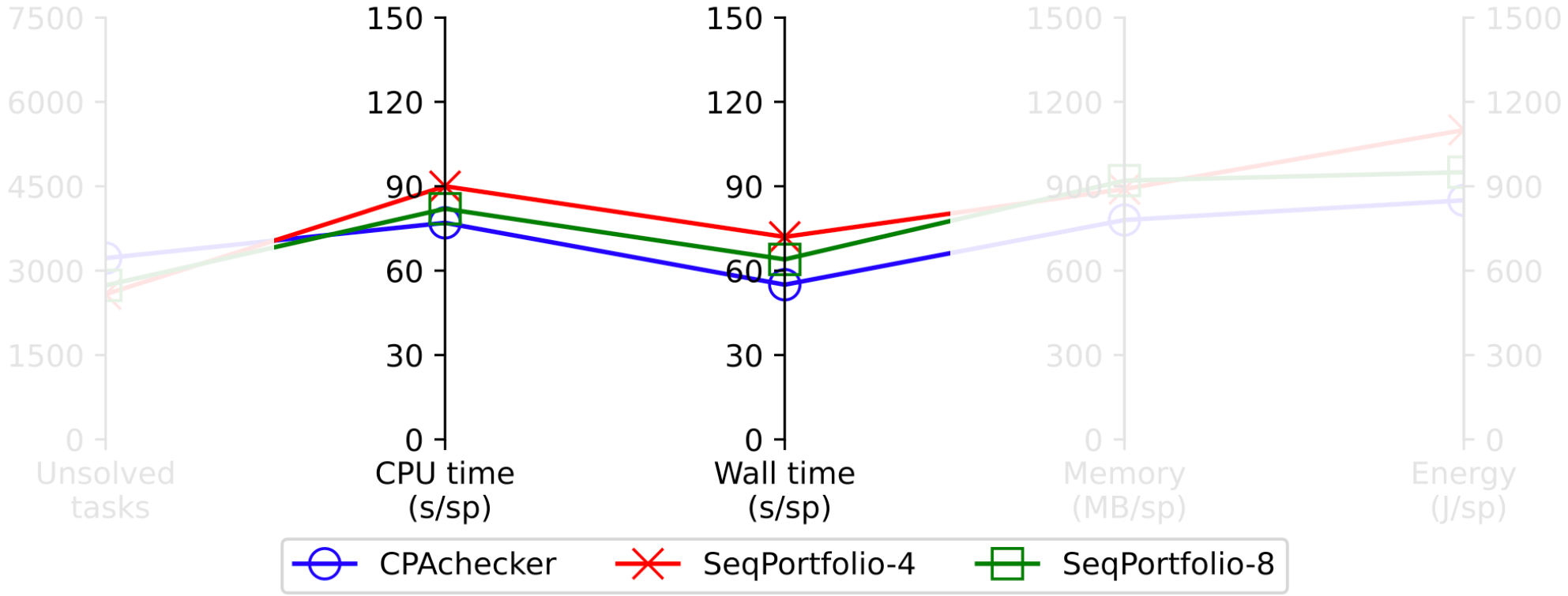
CPAchecker [7] ESBMC [30] Symbiotic [21] UAutomizer [31]

CBMC [40] Divine [41] Goblint [50] UTaipan [28]

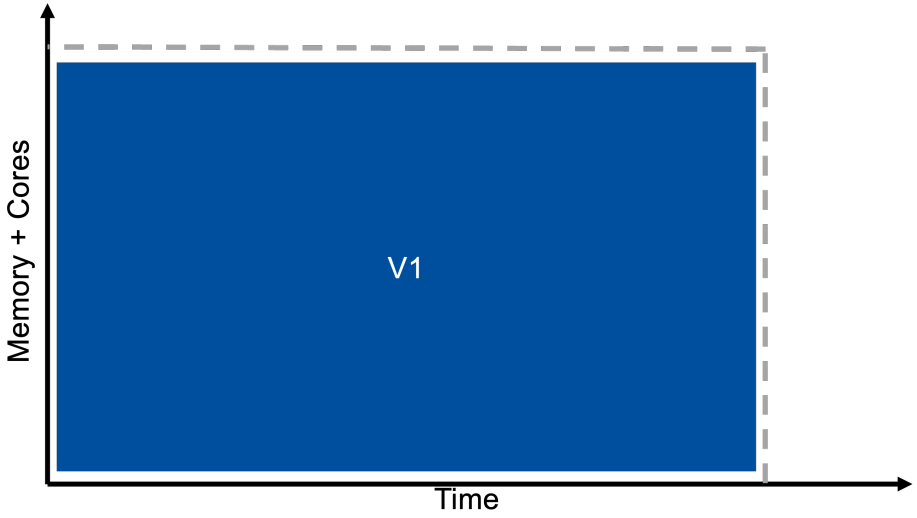
Sequential Portfolios come at a **negligible cost**

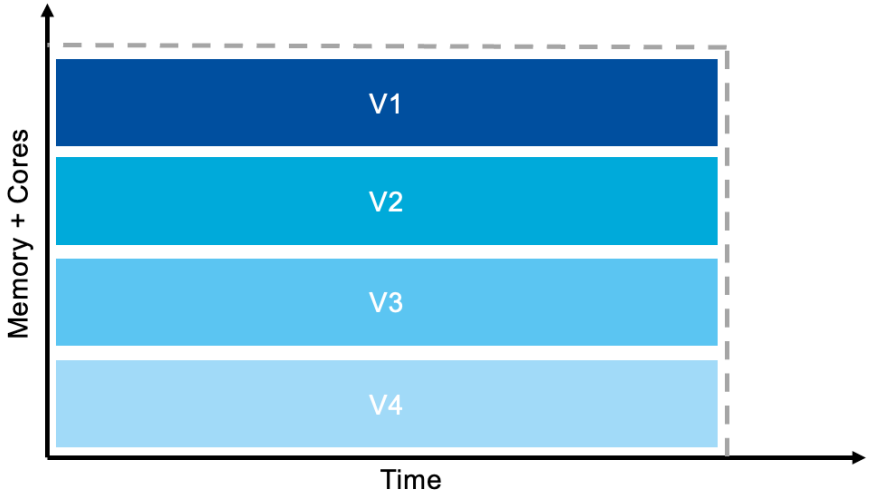


Sequential Portfolios come at a **negligible cost**



Can we improve a single verifier **by parallel portfolios**?





Carl von Ossietzky Universität Oldenburg

More verifiers can lead to more than **700** additional tasks solved

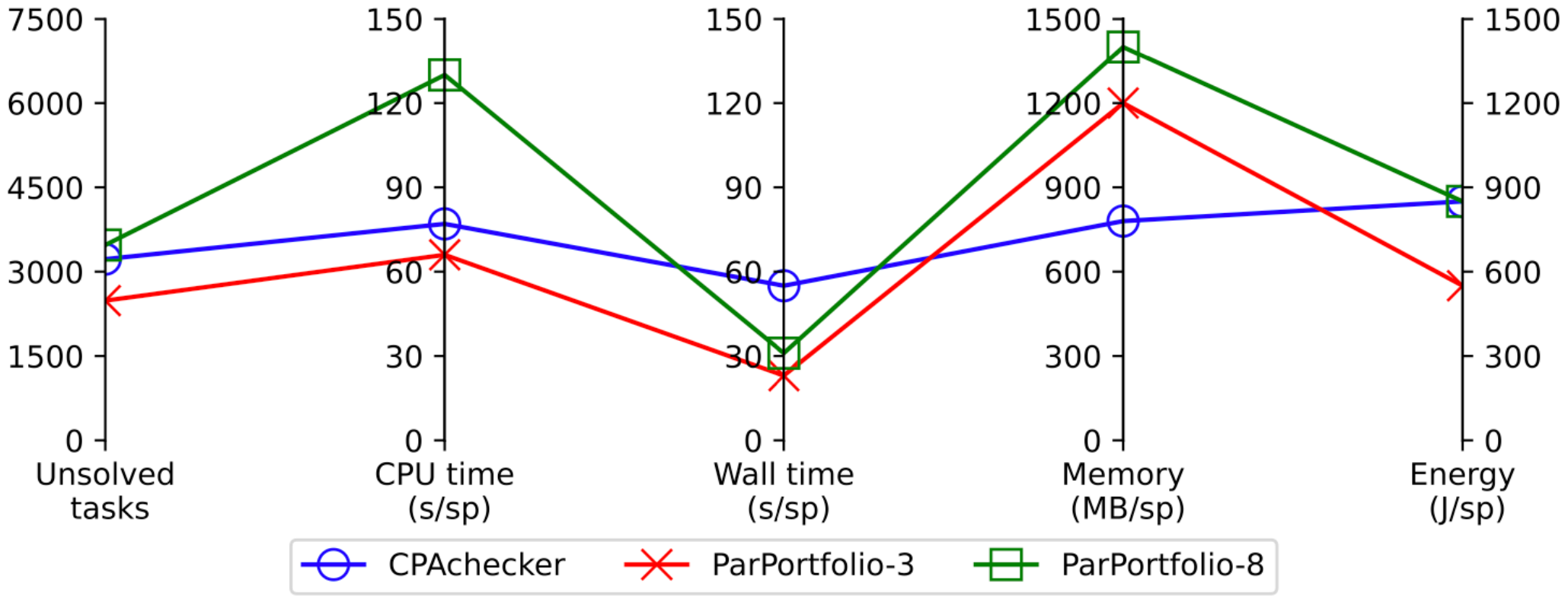
Verifier	CPAchecker	Parallel Portfolio of 2	Parallel Portfolio of 3	Parallel Portfolio of 4	Parallel Portfolio of 8
Score	9 040	8 969	**9 459**	8 952	7 547
Correct results	5 652	6 101	**6 363**	6 001	5 367
Correct proofs	3 516	3 780	**3 992**	3 639	3 236
Correct alarms	2 136	2 321	**2 371**	2 362	2 131
Wrong results	**8**	36	35	28	42
Wrong proofs	**0**	21	21	15	24
Wrong alarms	**8**	15	14	13	18

CPAchecker [7] ESBMC [30] Symbiotic [21]

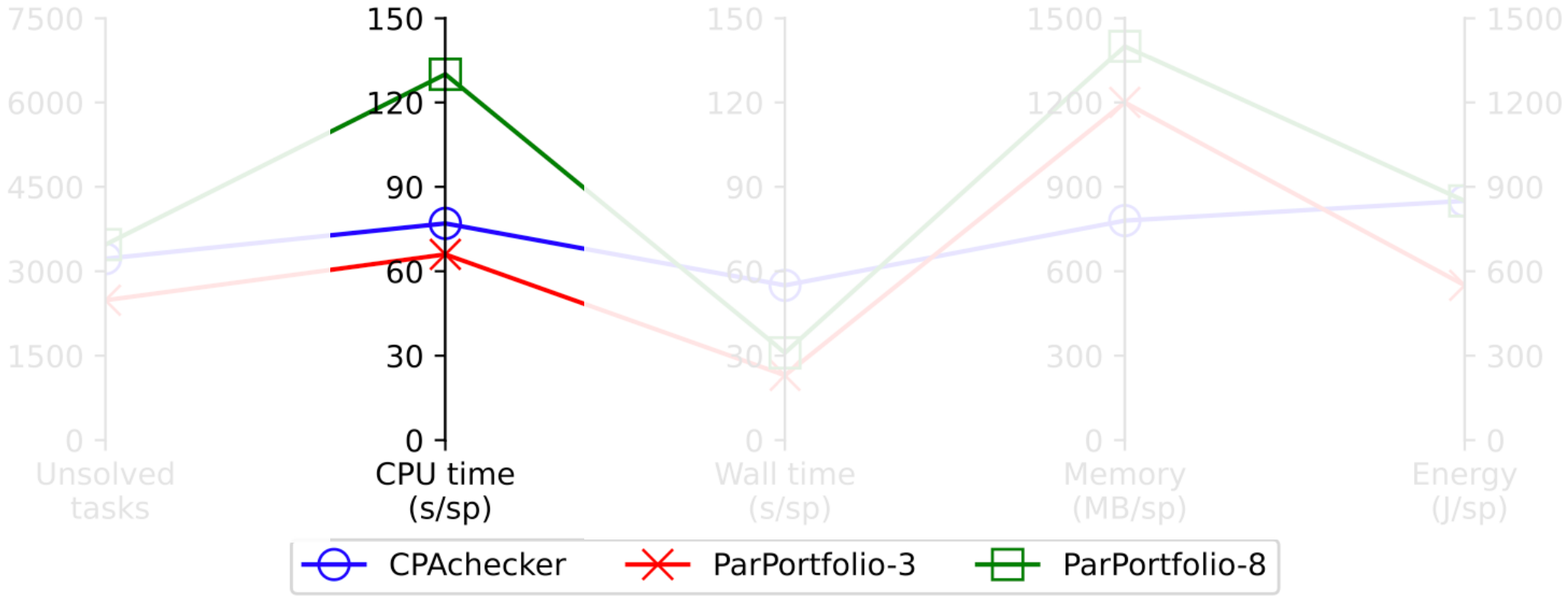
Limited by memory and cores

Verifier	CPAchecker	Parallel Portfolio of 2	Parallel Portfolio of 3	Parallel Portfolio of 4	Parallel Portfolio of 8
Score	9 040	8 969	9 459	8 952	7 547
Correct results	5 652	6 101	6 363	6 001	5 367
Correct proofs	3 516	3 780	3 992	3 639	3 236
Correct alarms	2 136	2 321	2 371	2 362	2 131
Wrong results	**8**	36	35	28	42
Wrong proofs	**0**	21	21	15	24
Wrong alarms	**8**	15	14	13	18

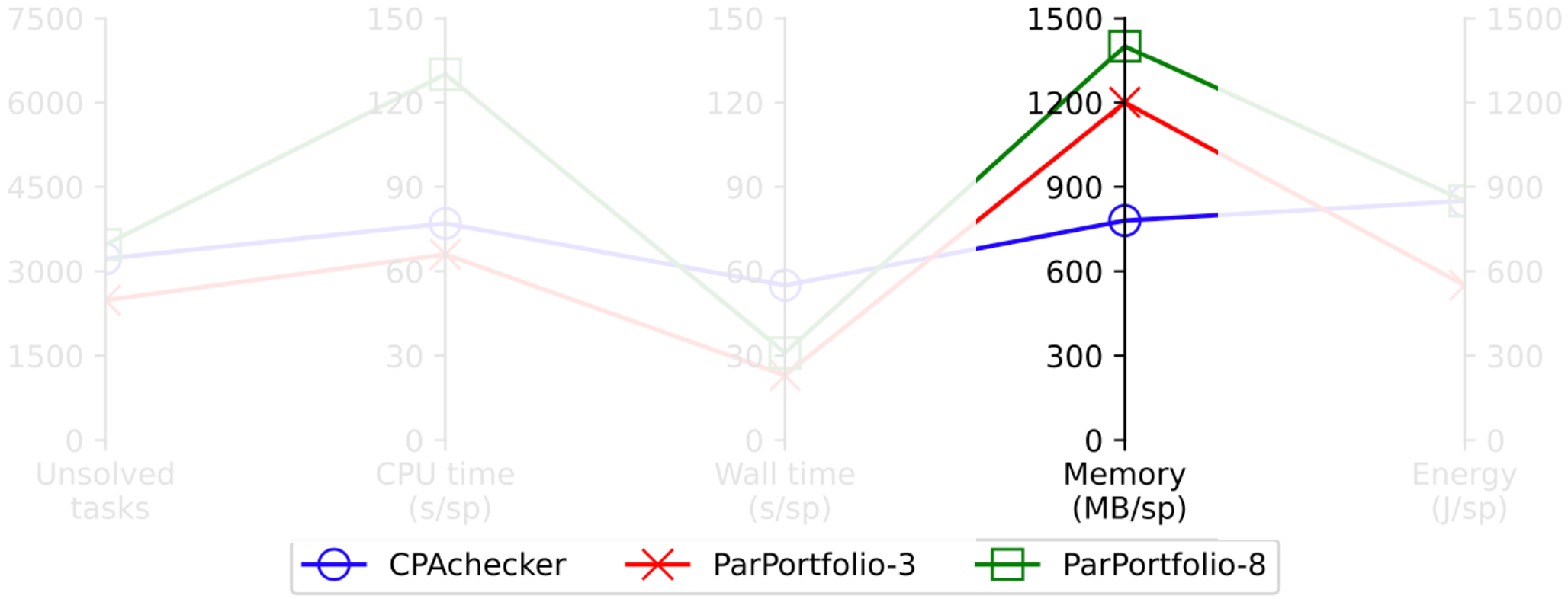
Parallel portfolios come at a **negligible cost**



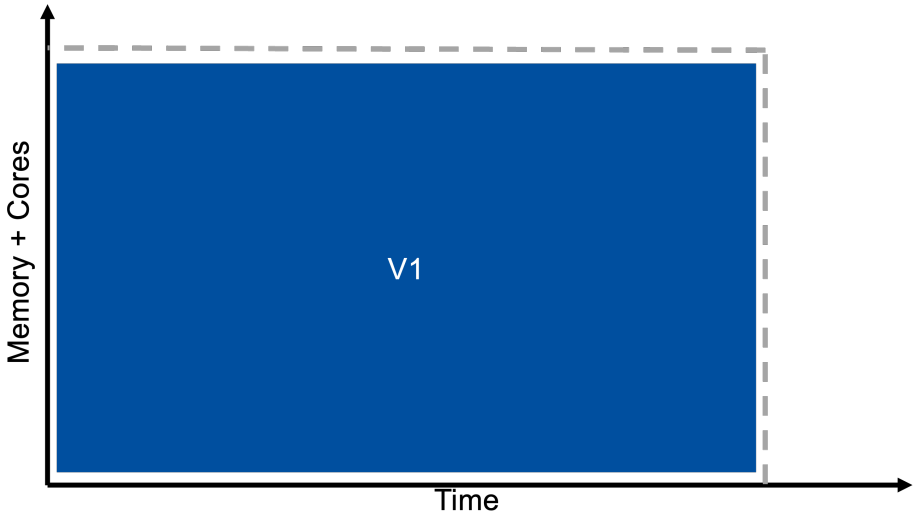
Parallel portfolios come at a **negligible cost**

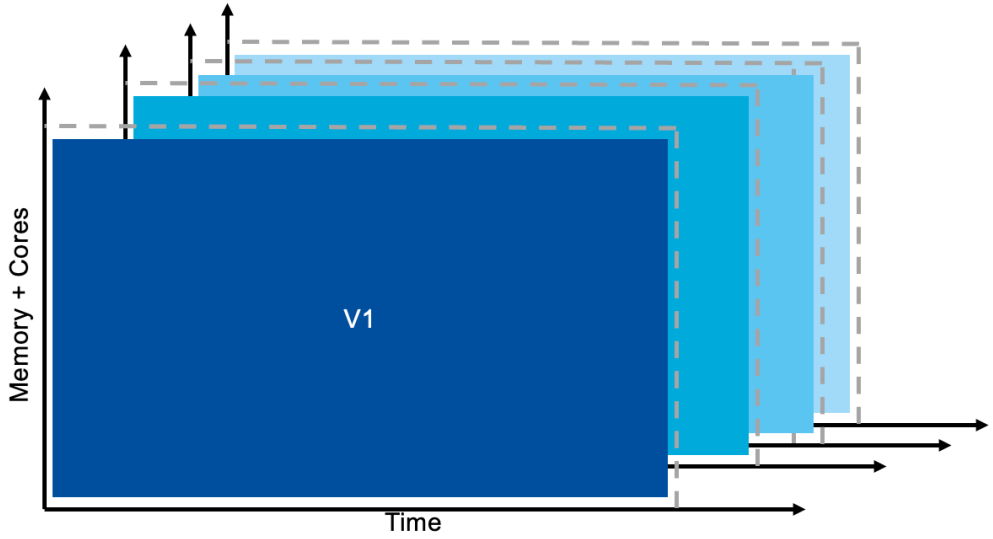


Parallel portfolios come at a **negligible cost**



Can we improve a single verifier **by algorithm selection**?





Carl von Ossietzky Universität Oldenburg

More verifiers lead to better performance

Verifier	CPAchecker	2	Algorithm Selection of 3	4	8
Score	9 040	9 226	9 689	9 816	**9 886**
Correct results	5 652	5 904	6 086	6 125	**6 214**
Correct proofs	3 516	3 658	3 843	3 867	**3 896**
Correct alarms	2 136	2 246	2 243	2 258	**2 318**
Wrong results	**8**	15	11	8	11
Wrong proofs	**0**	6	4	3	3
Wrong alarms	**8**	9	7	5	**8**

CPAchecker [7] ESBMC [30] Symbiotic [21] UAutomizer [31]

CBMC [40] Divine [41] Goblint [50] UTaipan [28]

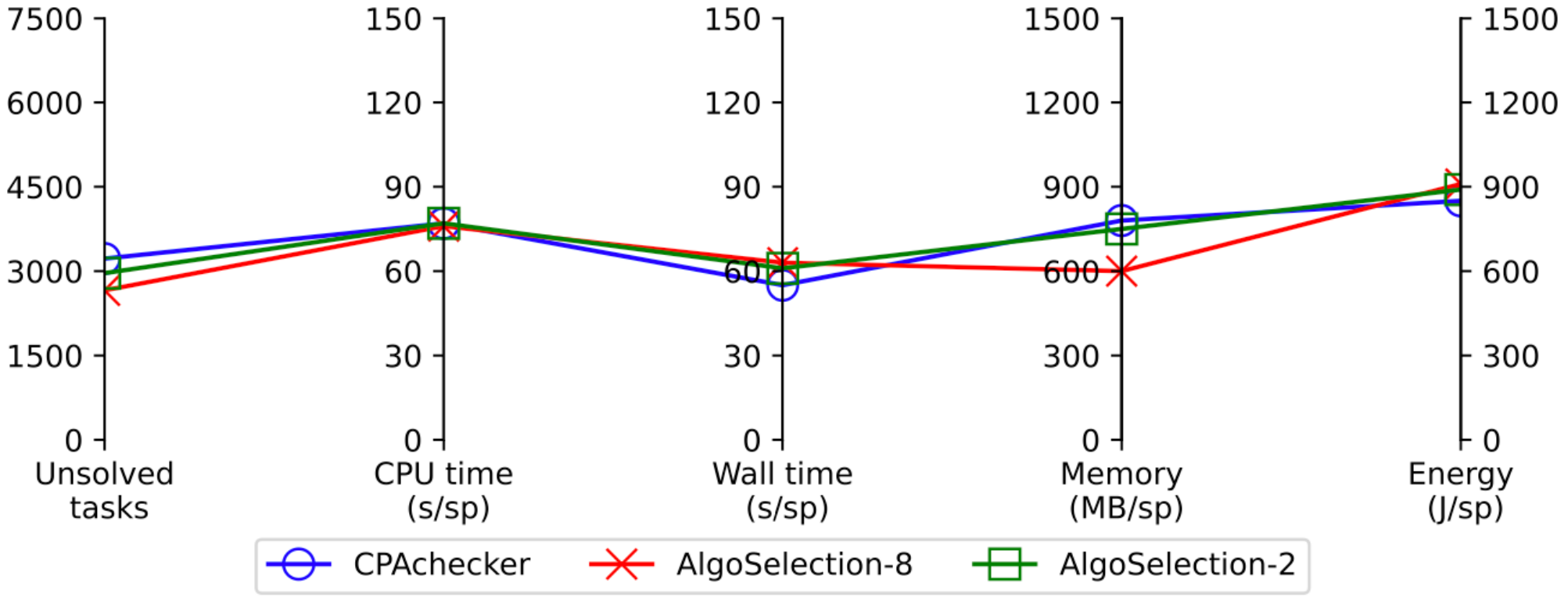
Clear **trend**

Verifier	CPAchecker	Algorithm Selection of 2	3	4	8
Score	9 040	9 226	9 689	9 816	**9 886**
Correct results	5 652	5 904	6 086	6 125	6 214
Correct proofs	3 516	3 658	3 843	3 867	3 896
Correct alarms	2 136	2 246	2 243	2 258	2 318
Wrong results	**8**	15	11	8	11
Wrong proofs	**0**	6	4	3	3
Wrong alarms	**8**	9	7	5	8

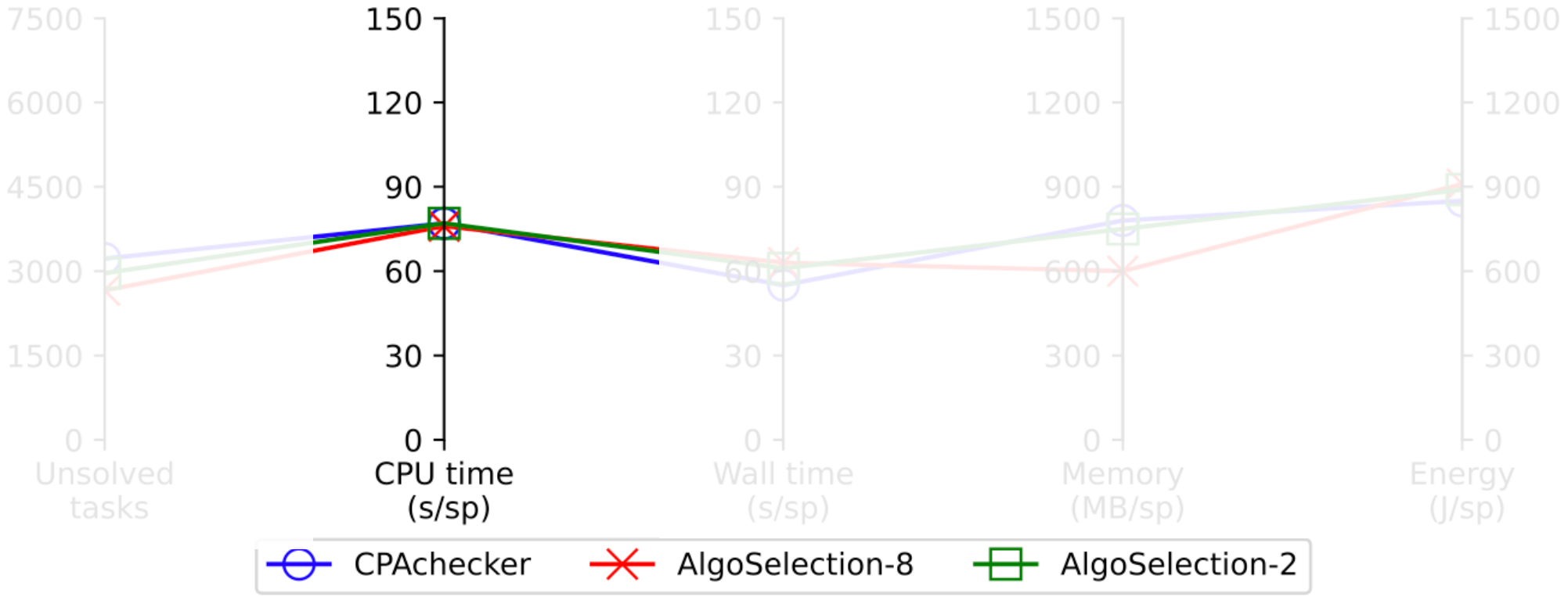
CPAchecker [7] ESBMC [30] Symbiotic [21] UAutomizer [31]

CBMC [40] Divine [41] Goblint [50] UTaipan [28]

Algorithm selection comes at a **negligible cost**

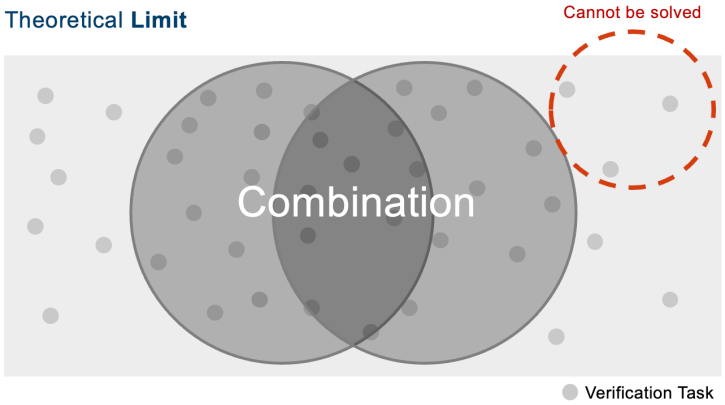


Algorithm selection comes at a **negligible cost**



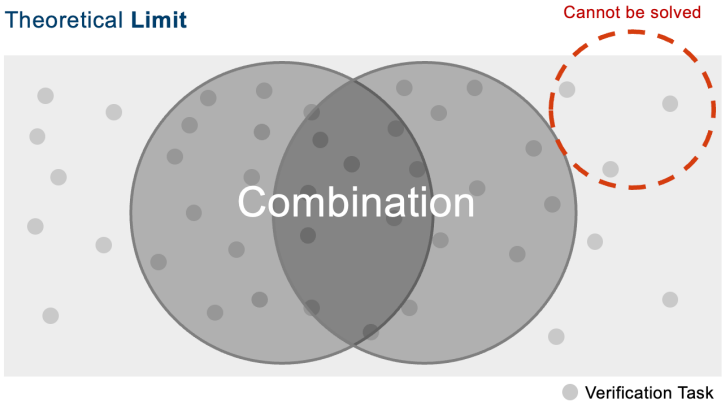
Future **Work**

Future **Work**

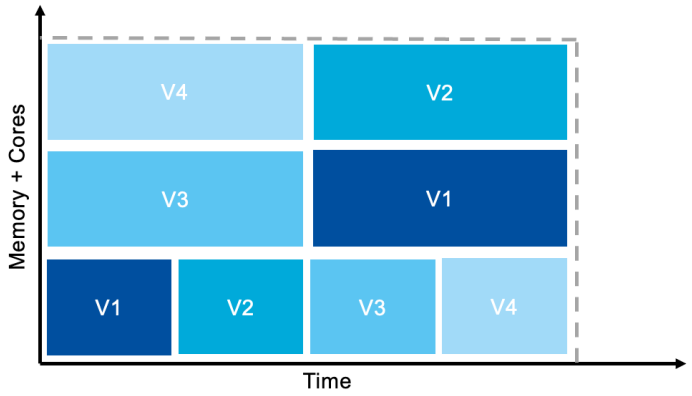


Theoretical Limit

Future **Work**

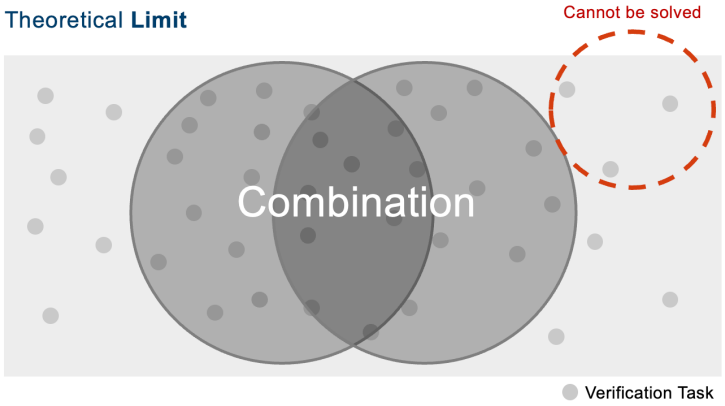


Theoretical Limit

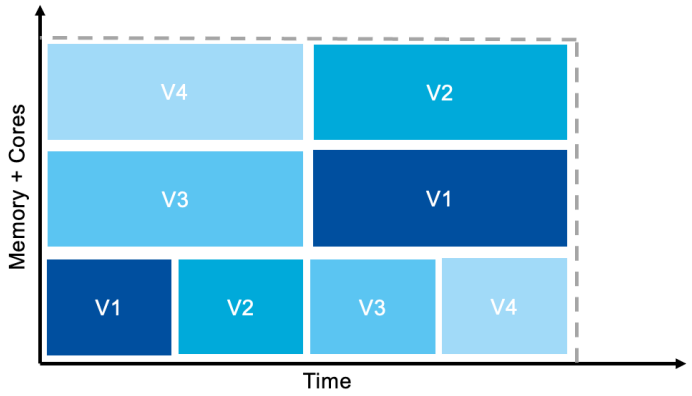


Hybrid approaches

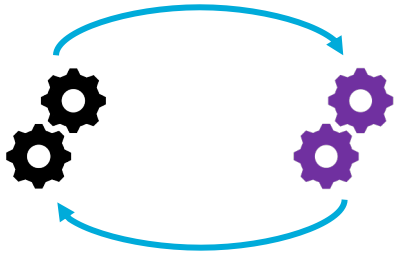
Future **Work**



Theoretical Limit



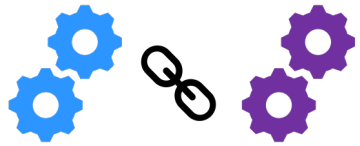
Hybrid approaches

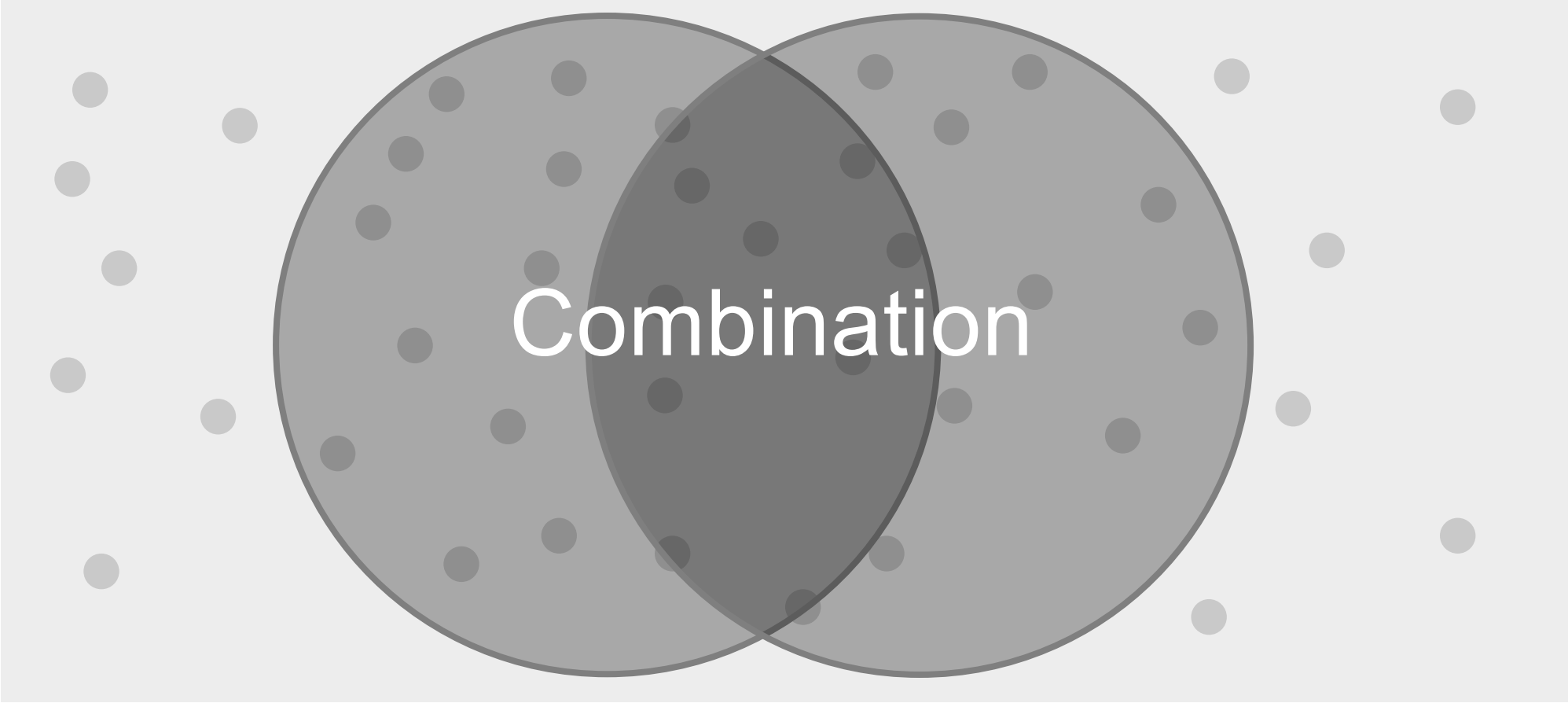


Cooperation

Takeaway

We can improve a single verifier **by combination!**





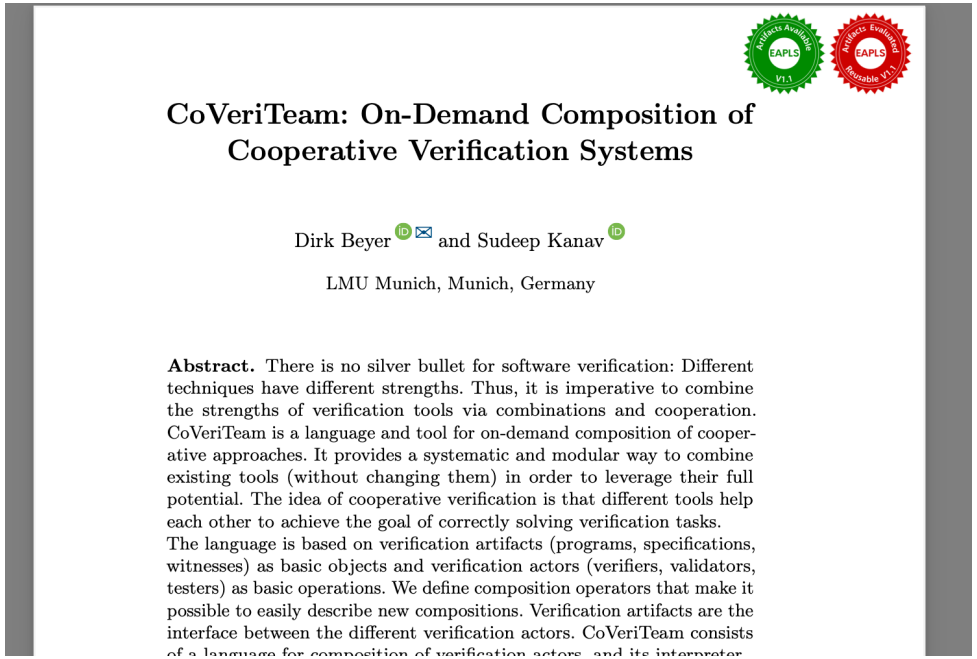
Takeaway

We

Learn more about **CoVeriTeam**!

TACAS Session

Wednesday,
at 15:30



CoVeriTeam: On-Demand Composition of Cooperative Verification Systems

Dirk Beyer and Sudeep Kanav

LMU Munich, Munich, Germany

Abstract. There is no silver bullet for software verification: Different techniques have different strengths. Thus, it is imperative to combine the strengths of verification tools via combinations and cooperation. CoVeriTeam is a language and tool for on-demand composition of cooperative approaches. It provides a systematic and modular way to combine existing tools (without changing them) in order to leverage their full potential. The idea of cooperative verification is that different tools help each other to achieve the goal of correctly solving verification tasks.
The language is based on verification artifacts (programs, specifications, witnesses) as basic objects and verification actors (verifiers, validators, testers) as basic operations. We define composition operators that make it possible to easily describe new compositions. Verification artifacts are the interface between the different verification actors. CoVeriTeam consists

Verification Task