2-query low-error PCP Composition

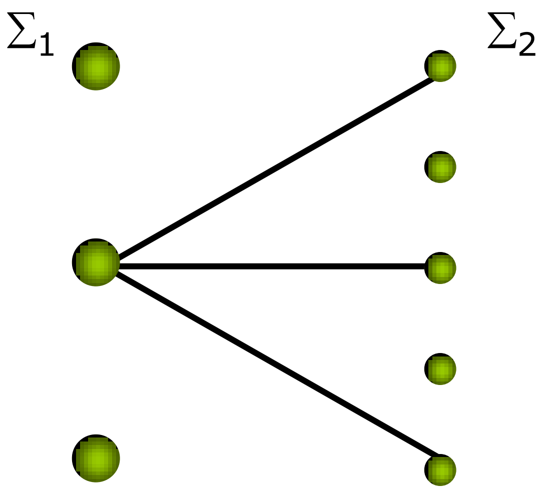
Prahladh Harsha
TIFR, Mumbai

Joint work with
Irit Dinur

Label Cover

Label Cover (LC)
G – Bipartite graph
Σ_1, Σ_2 – labels
(projection) constraint per edge
$c_{e:}\Sigma_1 \rightarrow \Sigma_2$

Edge e is satisfied if $c_e(\sigma_1)=\sigma_2$



Goal: Find an assignment to vertices that satisfies the most edges

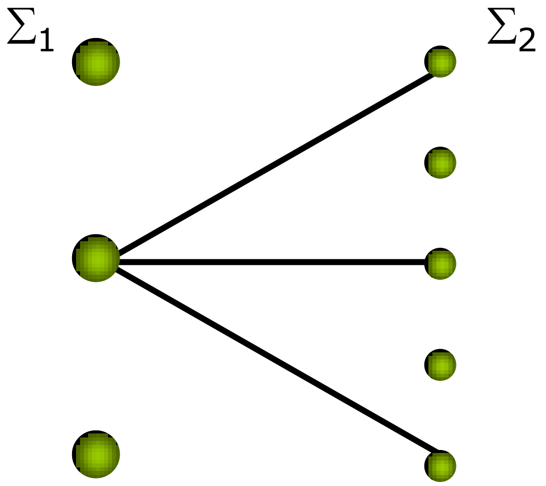
Gap(α,β)-LC: Distinguish between instances

- At least α fraction of constraints satisfied
- At most β fraction of constraints satisfied

Label Cover

Label Cover (LC)
G – Bipartite graph
Σ_1, Σ_2 – labels
(projection) constraint per edge
$c_e : \Sigma_1 \rightarrow \Sigma_2$

Edge e is satisfied if $c_e(\sigma_1)=\sigma_2$



PCP Theorem [Arora Safra '92,Arora Lund Motwani Sudan Szegedy '92]:

Gap(1,0.99999)-LC is NP-hard

Repetition Theorem [Raz '95]:

For every constant δ there exists alphabets Σ_1,Σ_2, Gap(1, δ)-LC is NP-hard

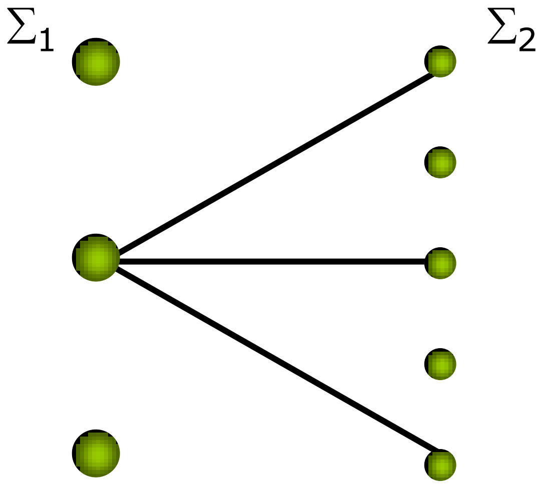
Consequence [Hastad '97]: MAX3SAT is hard to approximate within

(7/8 + ε) for every constant ε

Label Cover

Label Cover (LC)
G – Bipartite graph
Σ_1, Σ_2 – labels
(projection) constraint per edge
$c_e: \Sigma_1 \rightarrow \Sigma_2$

Edge e is satisfied if $c_e(\sigma_1)=\sigma_2$



PCP Theorem [Arora Safra '92,Arora Lund Motwani Sudan Szegedy '92]:

Gap(1,0.99999)-LC is NP-hard

Repetition Theorem [Raz '95]:

For every constant δ there exists alphabets Σ_1,Σ_2, Gap(1, δ)-LC is NP-hard

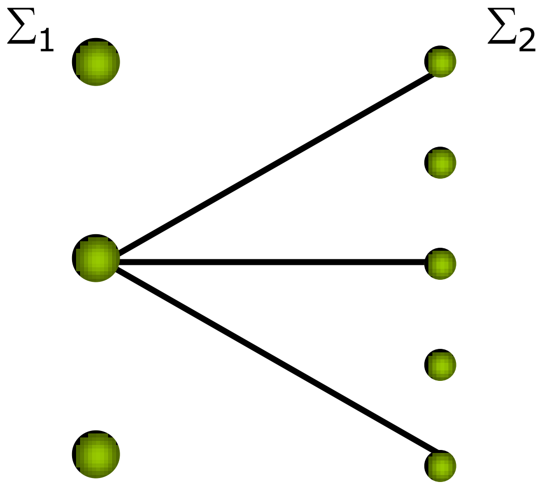
Consequence [Hastad '97]: MAX3SAT is hard to approximate within

(7/8 + ε) for every constant ε

Label Cover

<u>Label Cover (LC)</u>
G – Bipartite graph
Σ_1, Σ_2 – labels
(projection) constraint per edge
$c_{e:}\Sigma_1 \rightarrow \Sigma_2$

Edge e is satisfied if $c_e(\sigma_1)=\sigma_2$



Sub-constant error [Raz Safra '97, Arora Sudan '97]:

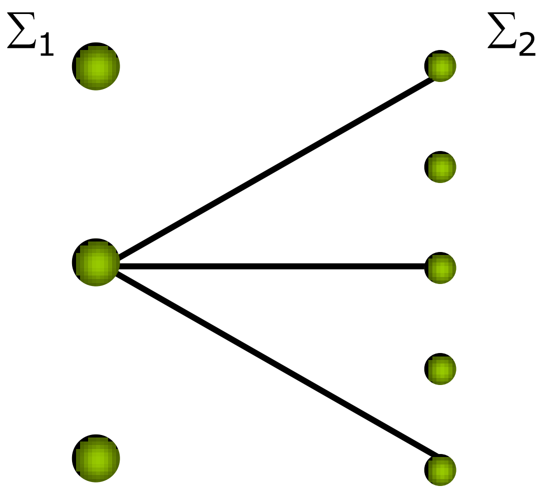
For every alphabet Σ and error $\delta = 1/\log|\Sigma|$, Gap(1, δ)-LC is NP-hard, provided $|\Sigma| > n^{\text{polylog } n}$

Caveat: Large Alphabet Size
Renders result "useless" for hardness results

Label Cover

<u>Label Cover (LC)</u>
G – Bipartite graph
Σ_1, Σ_2 – labels
(projection) constraint per edge
$c_e: \Sigma_1 \rightarrow \Sigma_2$

Edge e is satisfied if $c_e(\sigma_1)=\sigma_2$



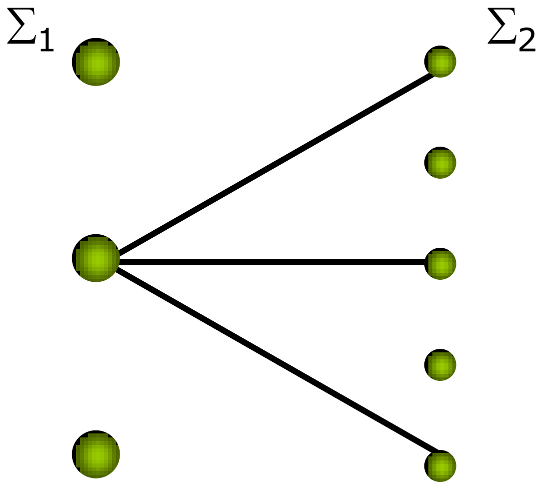
Sub-constant error [Moshkovitz Raz '08]:

For every alphabet Σ and error $\delta = 1/\log|\Sigma|$, Gap(1, δ)-LC is NP-hard

Label Cover

Label Cover (LC)
G – Bipartite graph
Σ_1, Σ_2 – labels
(projection) constraint per edge
$c_{e:}\Sigma_1 \rightarrow \Sigma_2$

Edge e is satisfied if $c_e(\sigma_1)=\sigma_2$



Sub-constant error [Moshkovitz Raz '08]:

For every alphabet Σ and error $\delta=1/\log|\Sigma|$, Gap(1, δ)-LC is NP-hard

[even under nearly linear sized reductions]

Core of [MR'08]

- Alphabet Reduction:
  - Label Cover instance with large alphabet size
  - ↓
  - Label Cover instance with small alphabet size
  - Difficulty:
    - Reduction must not affect soundness error δ
    - Target instance must be label cover instance
  - Intricate and fairly involved construction

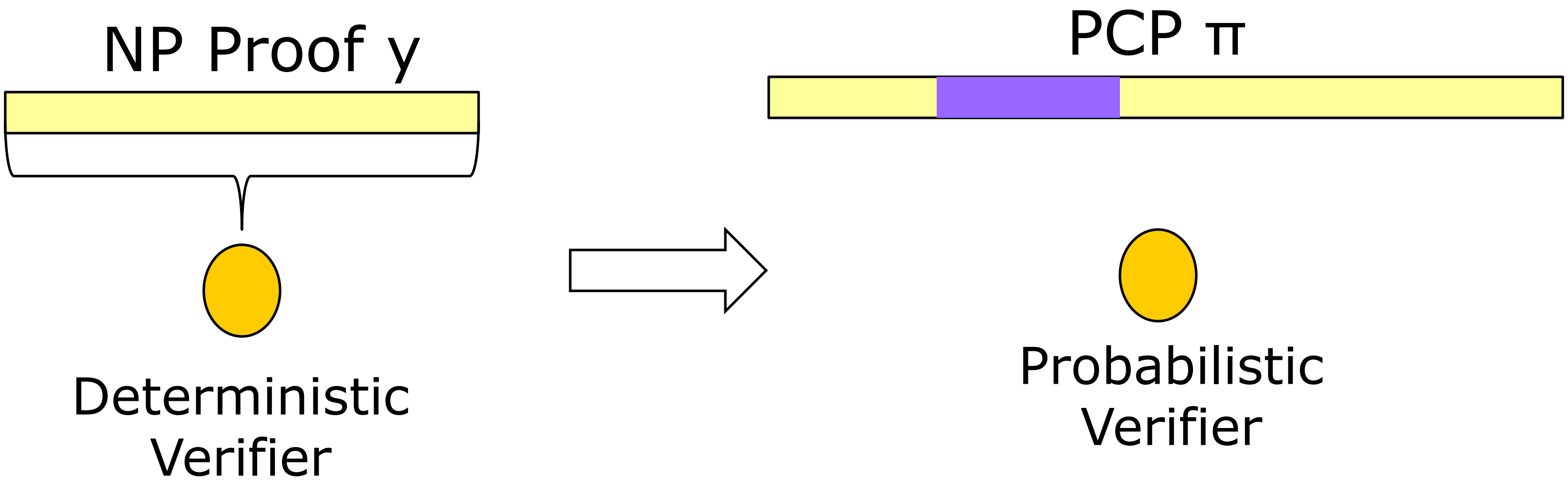
Alphabet Reduction [Dinur H. '09]

- A new Alphabet Reduction Technique
  (aka Composition Theorem)
  - Large alphabet to small alphabet
    - (without affecting soundness δ too much)
  - Generic composition, works with any label cover instance
  - Gives simpler proof of [MR'08]

Probabilistically Checkable Proofs (PCPs)

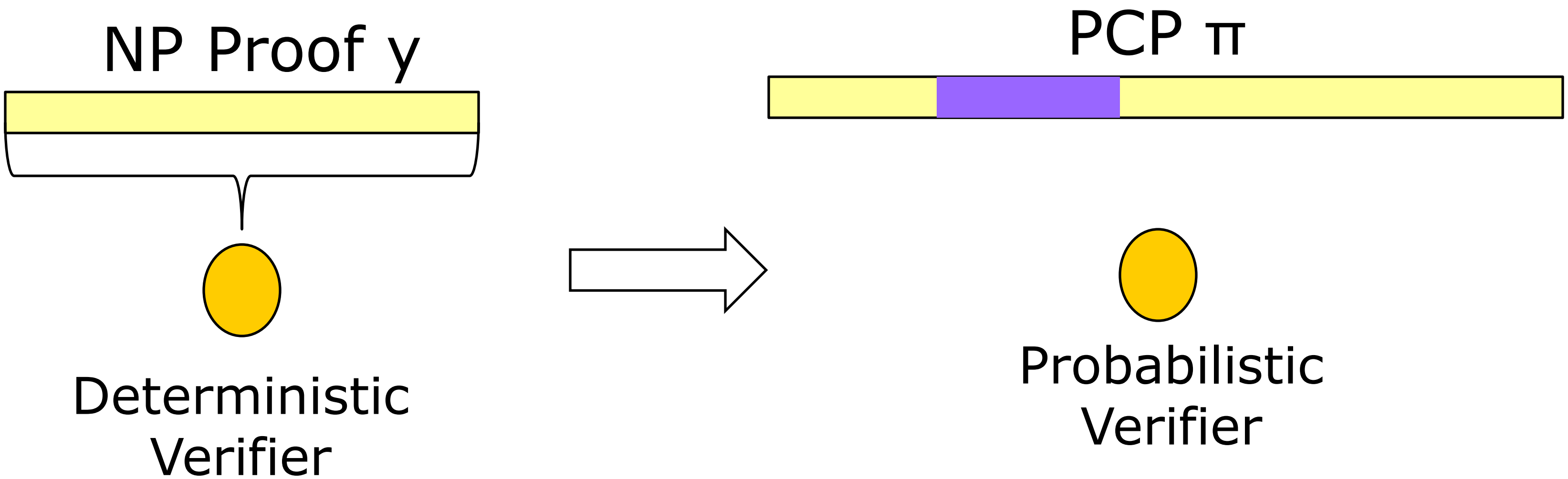
Probabilistically Checkable Proofs

- PCP Theorem: characterization of NP



Probabilistically Checkable Proofs

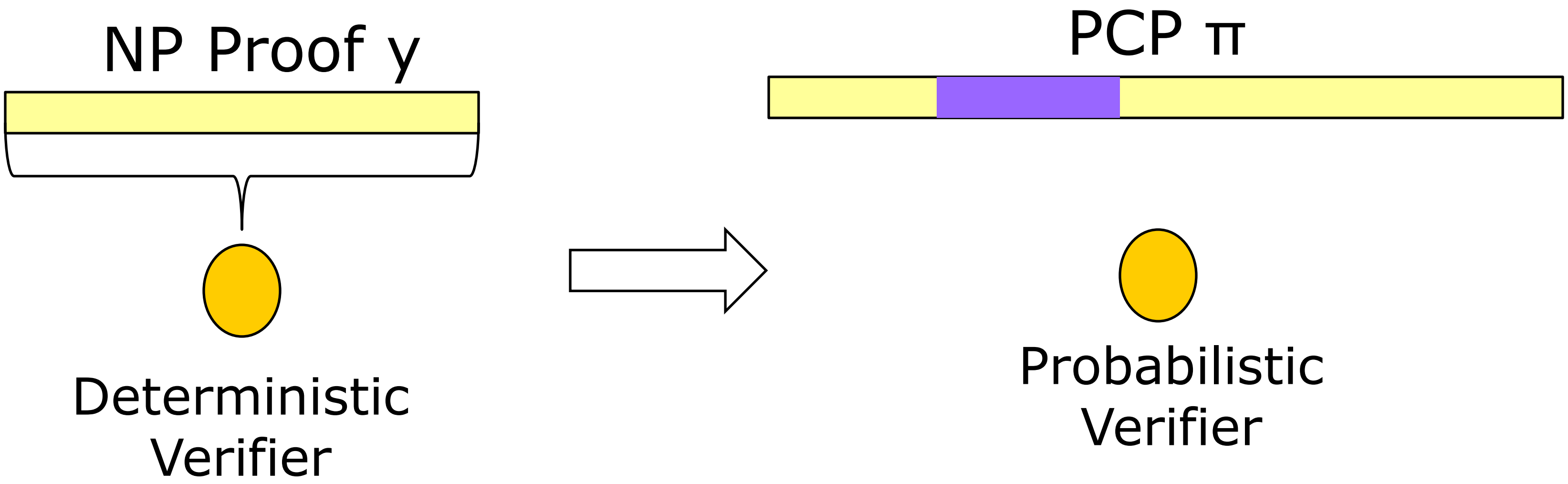
- PCP Theorem: characterization of NP



PCP π – locally testable encoding of the NP proof y

Probabilistically Checkable Proofs

- PCP Theorem: characterization of NP



Completeness:

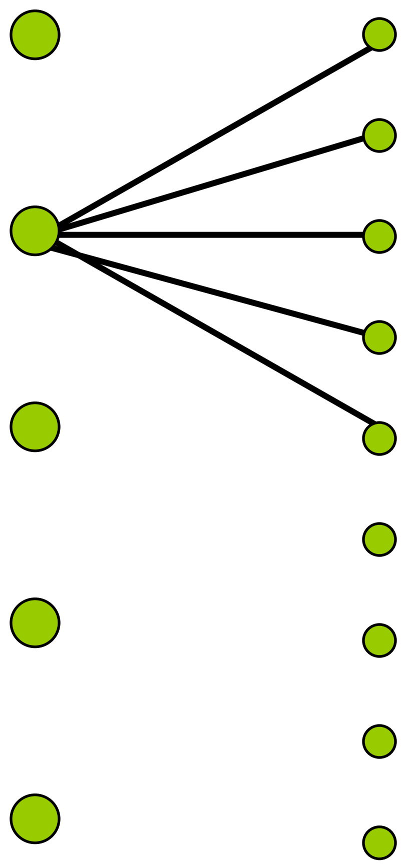
If $\varphi \in$ SAT, there is a PCP π such all local views are accepting

Soundness:

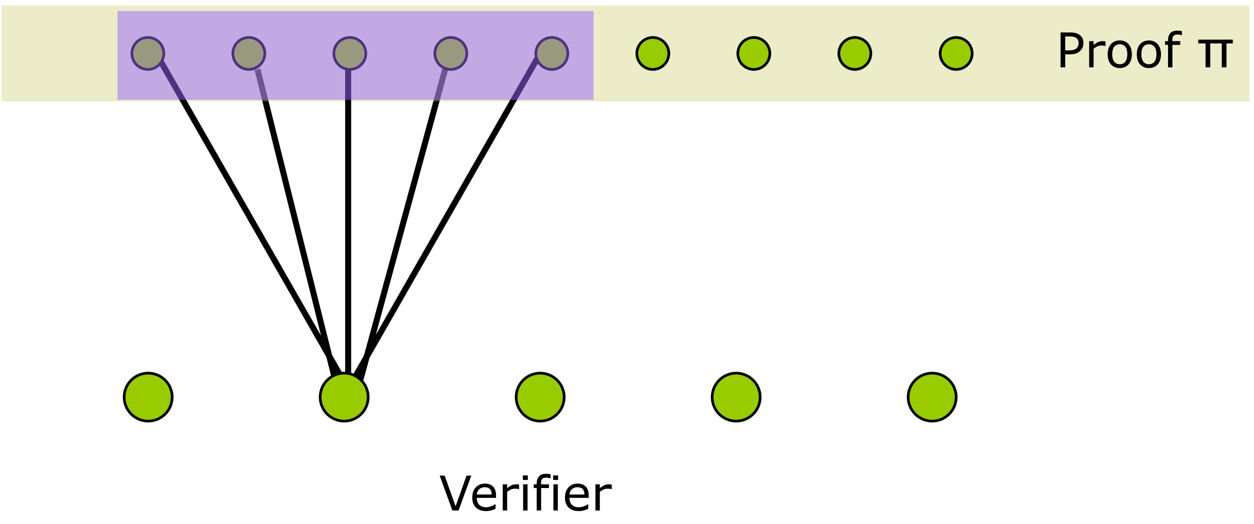
If $\varphi \notin L$, then for all PCPs π most local views are rejecting

PCPs to Label Cover?

Label Cover ←→ PCPs



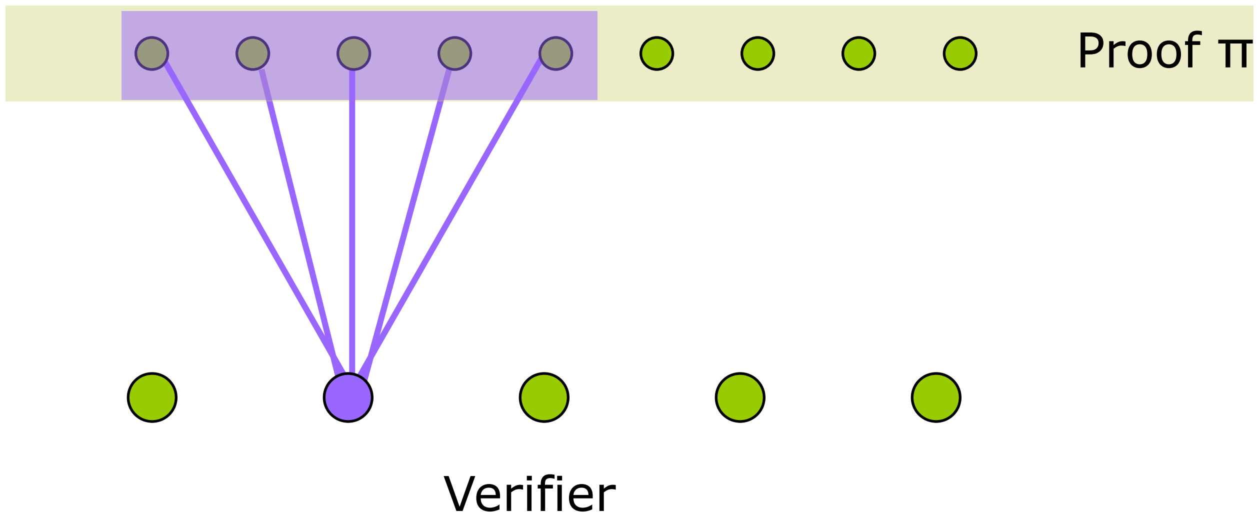
Label Cover ←→ PCPs



Verifier

1. Selects a random “big” vertex **u**
2. Reads entire neighborhood of **u**
3. Accepts iff there is a value for **u** that would cause all edge constraints to accept.

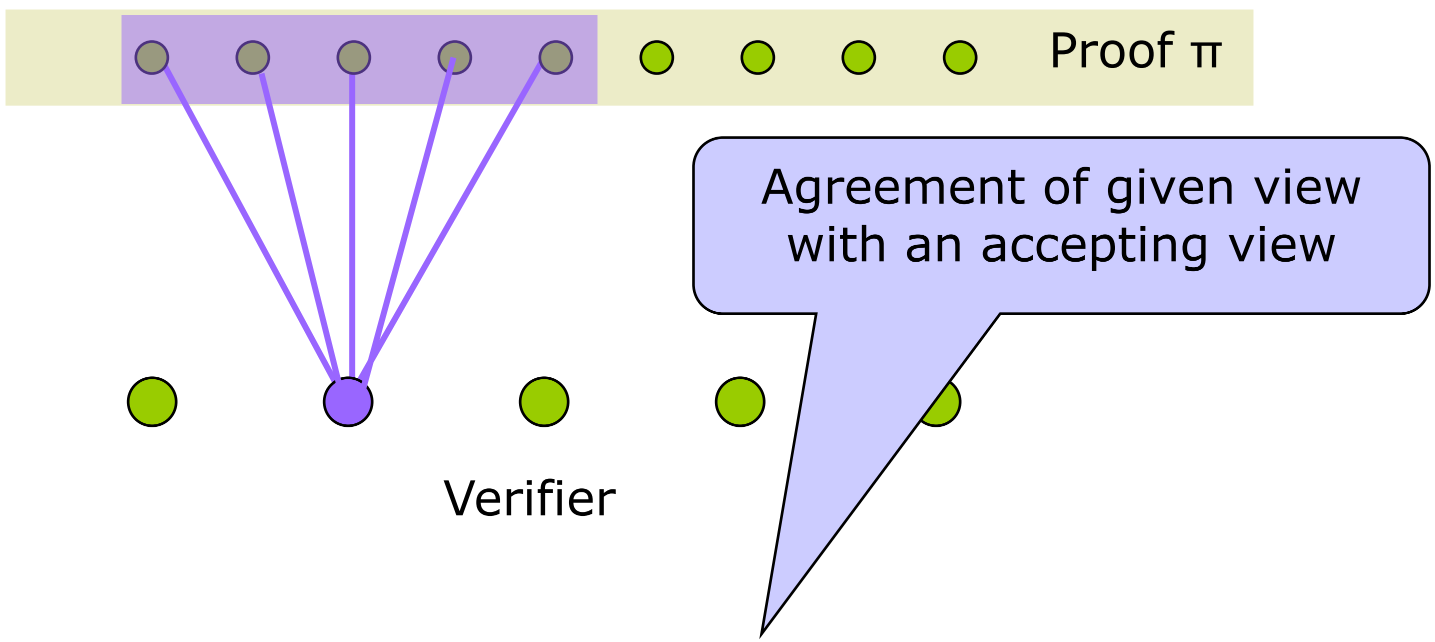
Label Cover ←→ PCPs



YES instances – all views are "happy"

NO instances – average view is very "unhappy", i.e. view from a random window is at most δ-close to a satisfying view.

Label Cover ←→ PCPs



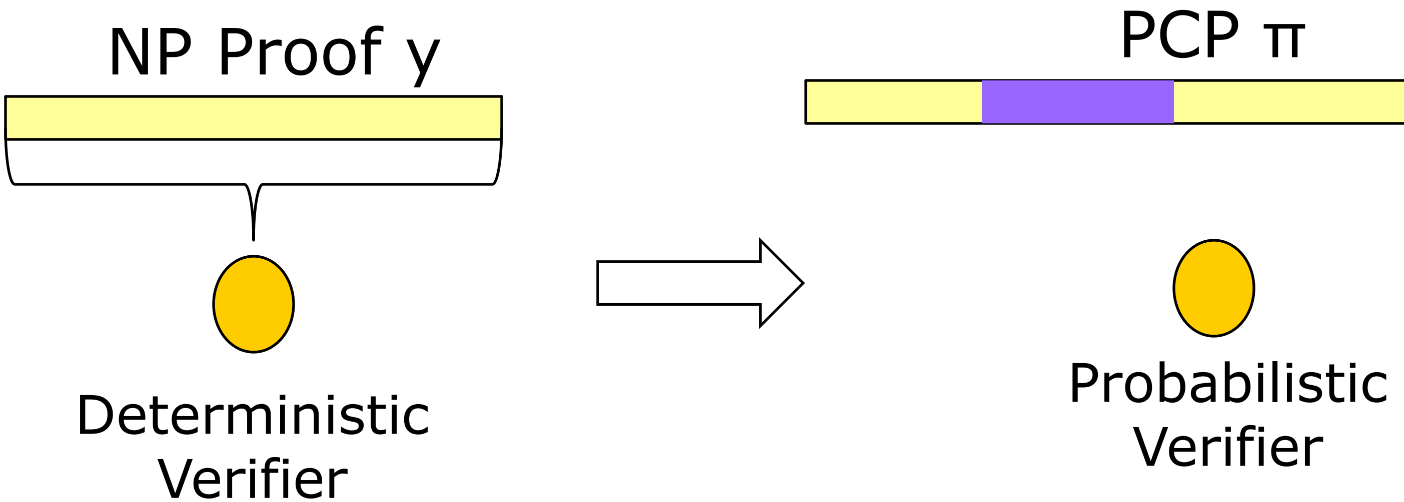
1) YES: $\exists\, \pi,$ E_r [agreement(π(r))]=1

2) NO: $\forall \pi,$ E_r [agreement(π(r))] < δ

- Robust soundness implies regular soundness
- But not vice versa

PCPs

- PCP Theorem: characterization of NP



Completeness:

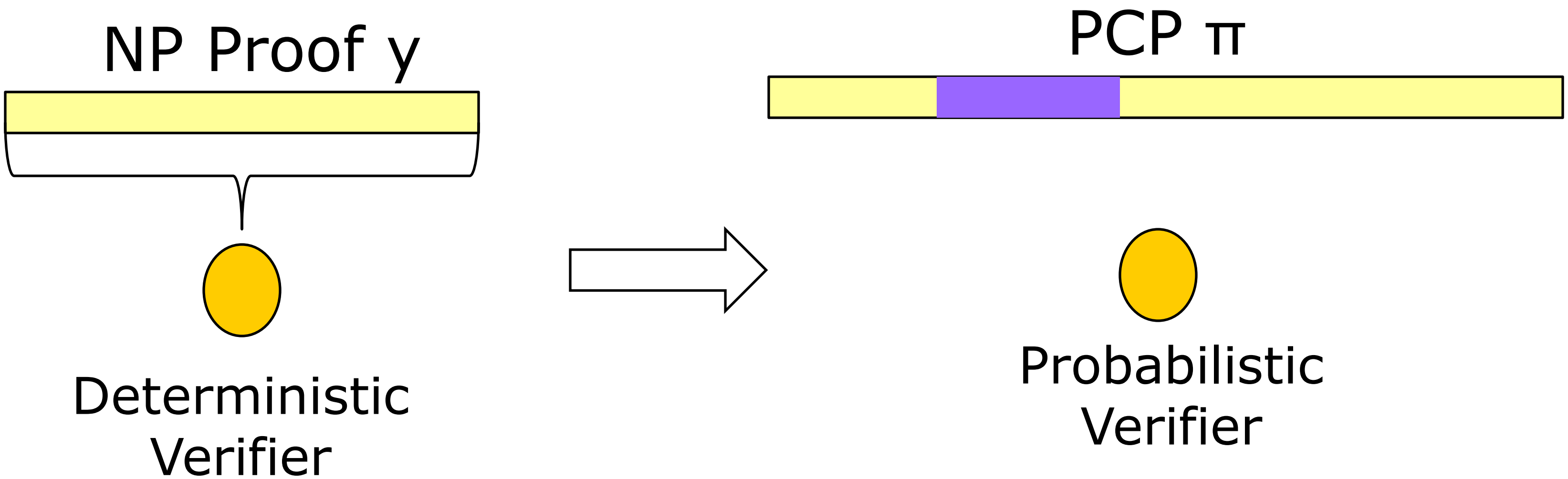
If $\varphi \in$ SAT, there is a PCP π such all local views are accepting

Soundness:

If $\varphi \notin$ L, then for all PCPs π most local views are rejecting

Robust PCPs [BGHSV '04]

- PCP Theorem: characterization of NP



Completeness:

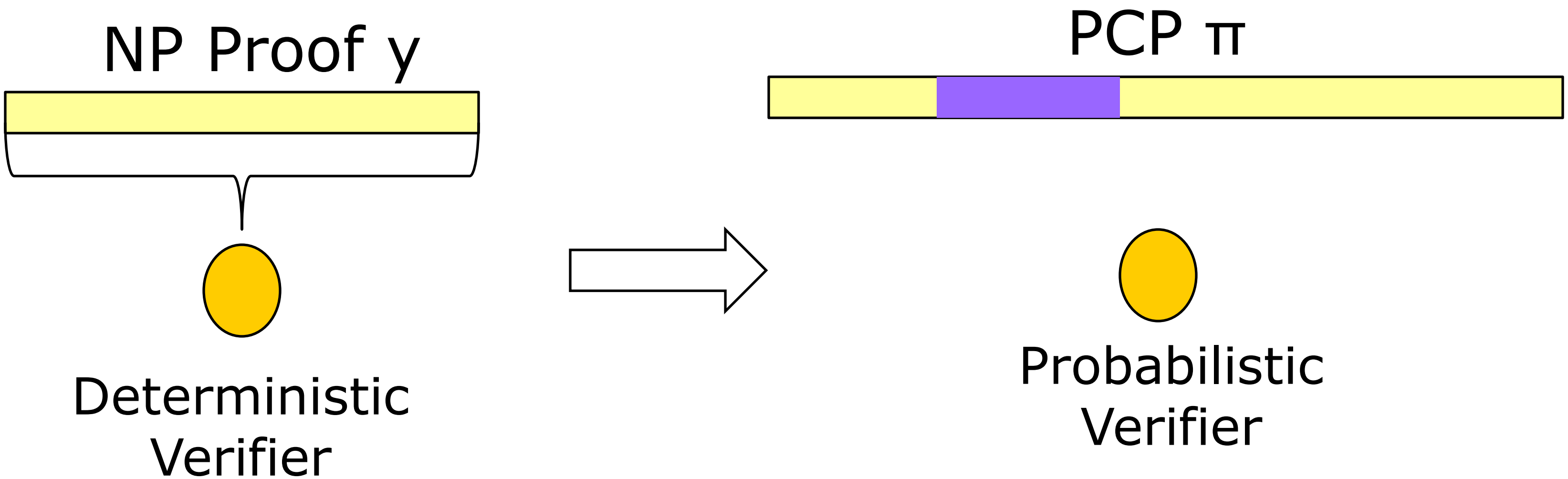
If $\varphi \in$ SAT, there is a PCP π such all local views are accepting

Robust Soundness:

If $\varphi \notin L$, then for all PCPs π most local views are far from accepting

Robust PCPs [BGHSV '04]

- PCP Theorem: characterization of NP



Completeness:

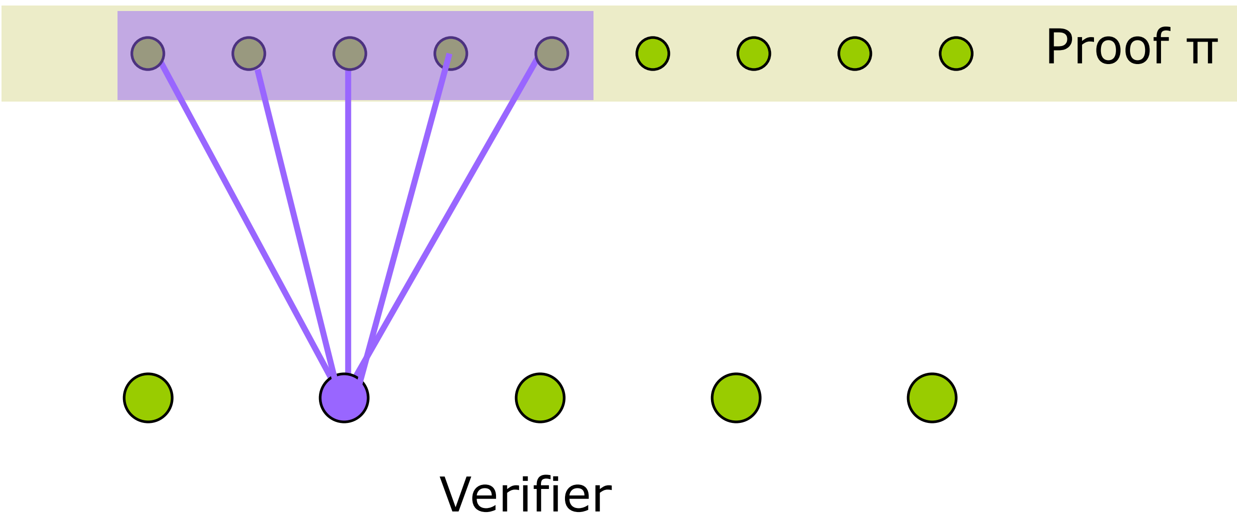
If $\varphi \in$ SAT, there is a PCP π such all local views are accepting

Robust Soundness:

If $\varphi \notin$ L, then for all PCPs π most local views are far from accepting

robust soundness =E[agreement( )] < δ

Label Cover ←→ Robust PCPs



- This transformation is "invertible"
- $|\Sigma_1|$ corresponds to the number of accepting configurations, which is bounded by exp(window size) = exp(# queries)

Label Cover ←→ Robust PCPs

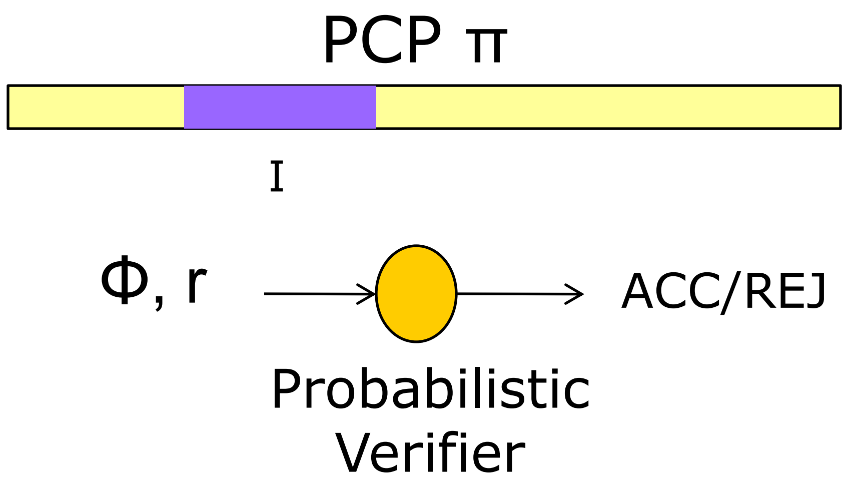
- Equivalence
  - PCP for SAT with robust soundness δ
  - There is a reduction from SAT to Gap(1, δ)-LC
- In this equivalence,
  - alphabet size of Label Cover = #queries of PCP

Goal Restated

- Label Cover:
  - Reduce alphabet size

- Robust PCP:
  - Reduce #queries

Reducing # queries



Verifier's Actions

1. Read inputs Φ, r

1. Compute local window I and local predicate f

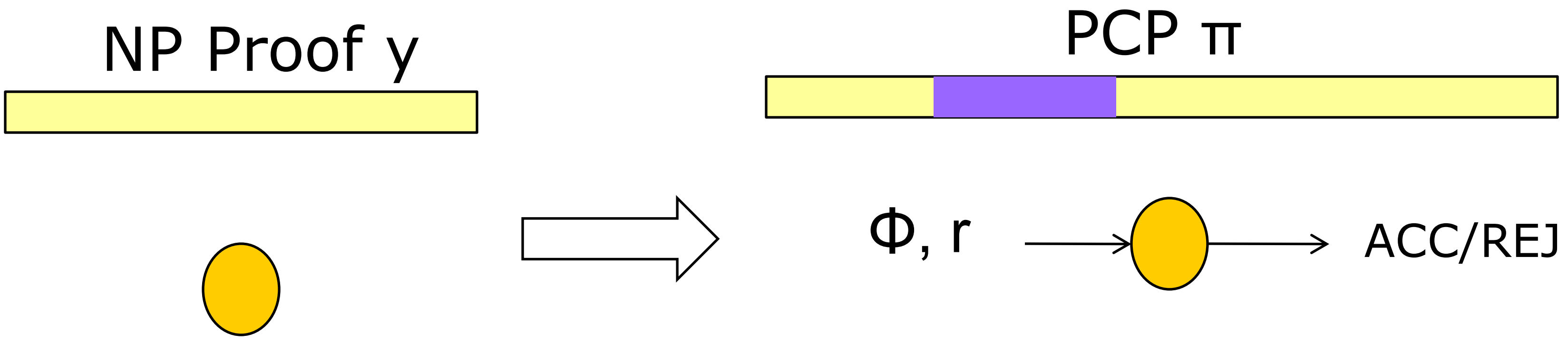
Idea: Compose!!
[ala composition of AS'92]

Use "Inner" PCP Verifier to check if local window satisfies local predicate

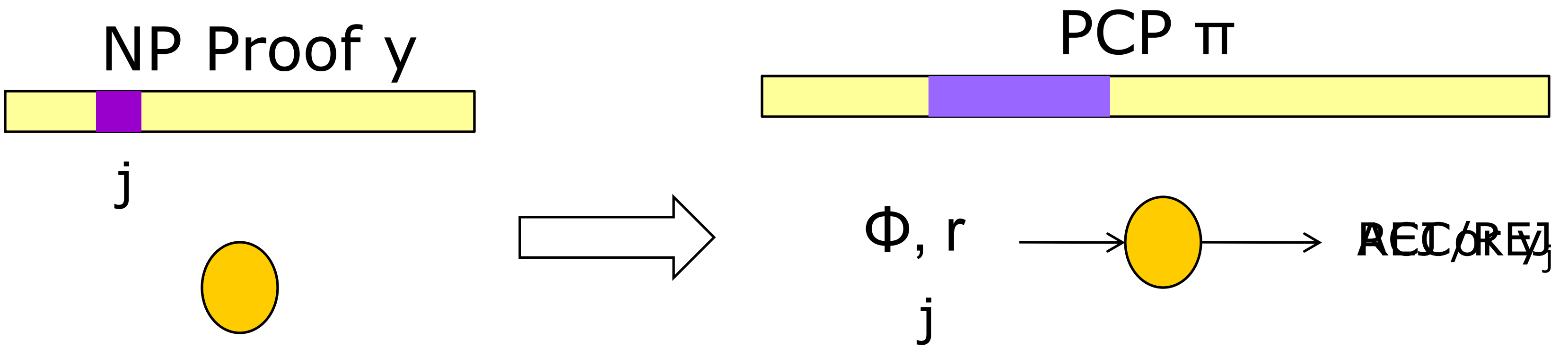
Consistency Issue: Inner verifier not only needs to check local predicate is satisfiable (easy), but also that is satisfiable by local window

Resolve Consistency using PCPs that can decode!!

Decodable PCPs



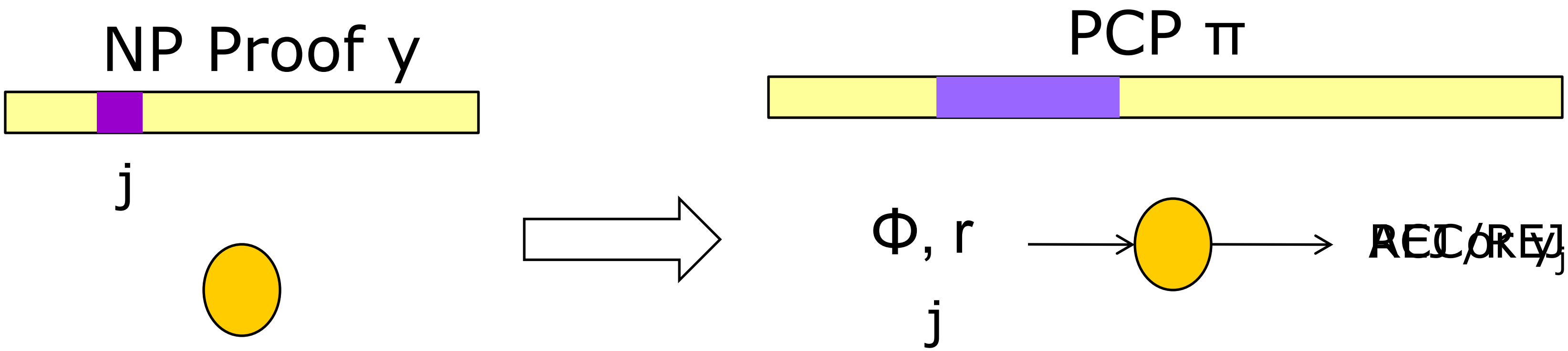
Decodable PCPs



Decodable PCP (dPCP) – encoding of NP proof

- locally testable
- locally decodable

Decodable PCPs



Completeness:

For every NP proof y, there is a dPCP π such that

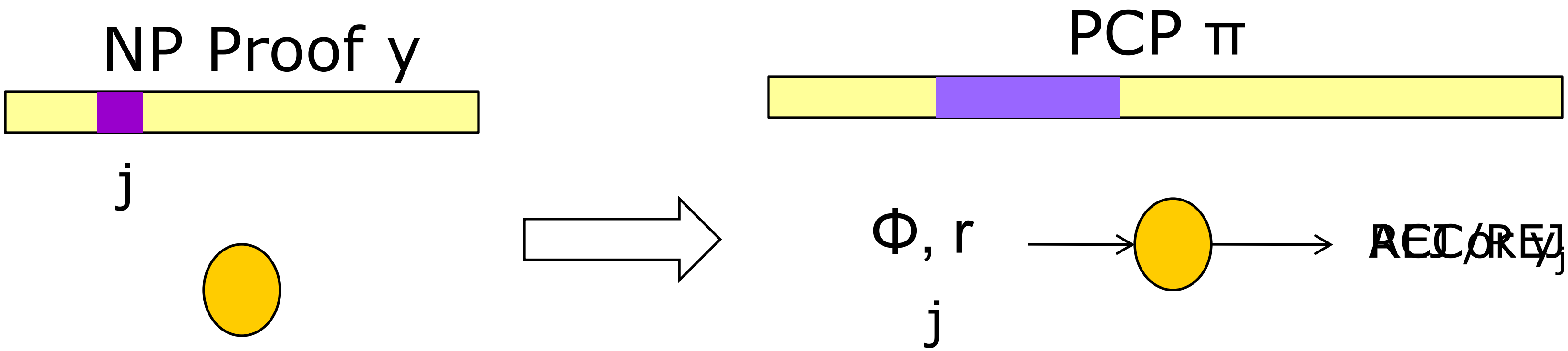
Pr[Verifier decodes correctly] = 1

Soundness:

For every dPCP π, there is at most a NP proof y

Pr[Verifier's output inconsistent with y] < δ

Decodable PCPs



Completeness:

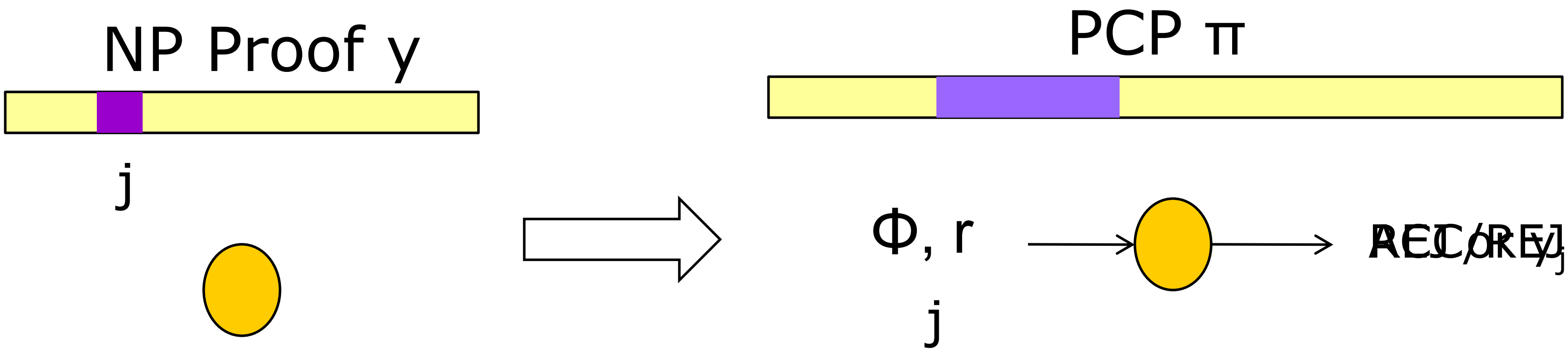
For every NP proof y, there is a dPCP $\pi=\pi(y)$ such that $\mathrm{Prob}_{i,r}[\ f(\pi_I)=y_i\ ] = 1$

Soundness:

For every dPCP π, there is at most a NP proof y

Pr[Verifier's output inconsistent with y] < δ

Decodable PCPs



Completeness:

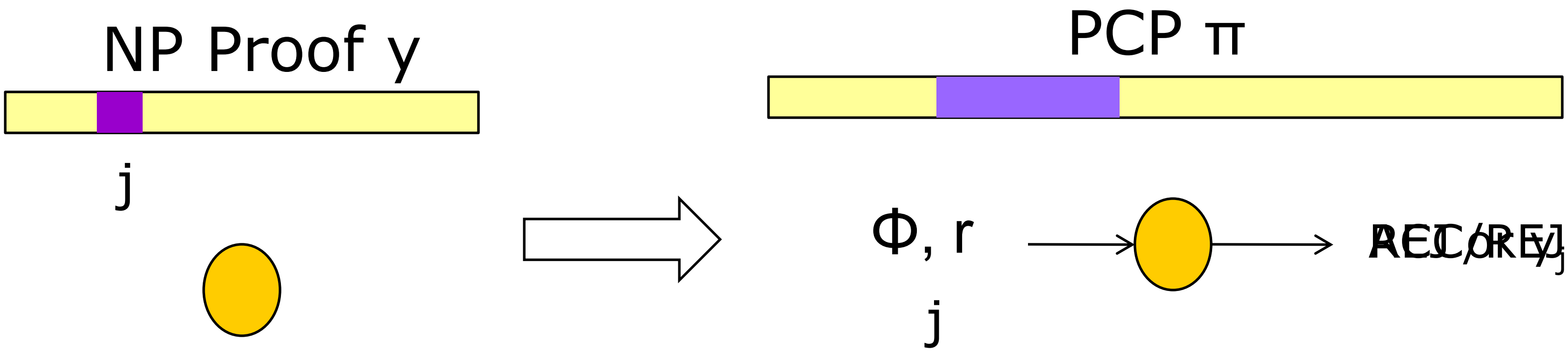
For every NP proof y, there is a dPCP $\pi=\pi(y)$ such that $\text{Prob}_{i,r}[\ f(\pi_I)=y_i\ ] = 1$

Soundness:

For every dPCP π, there is at most a NP proof y

$\text{Prob}_{i,r}[\ f(\pi_I) \notin \{(y)_i\} \cup \{\text{reject}\}\ ] < \delta$

Decodable PCPs



Completeness:

For every NP proof y, there is a dPCP $\pi=\pi(y)$ such that $\mathrm{Prob}_{i,r}[\ f(\pi_I)=y_i\ ] = 1$

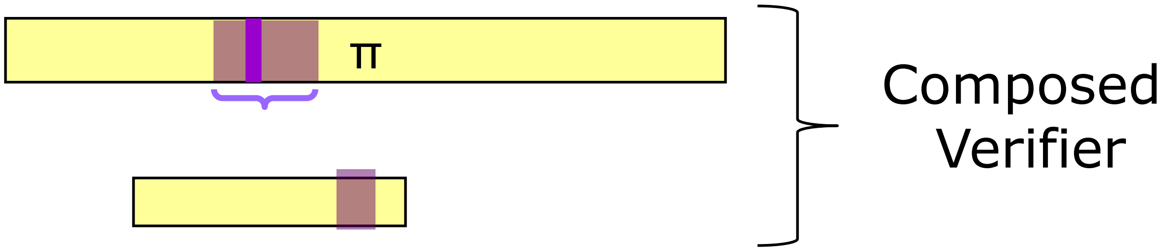
Soundness:

For every dPCP π, there is a short list of NP proofs $y^1,..,y^L$, $\mathrm{Prob}_{i,r}[\ f(\pi_I) \notin \{(y^j)_i\} \cup \{\text{reject}\}\ ] < \delta$

Composition with dPCPs

- Composition:

1. The verifier first computes local window I and local predicate
2. Instead of checking local window satisfies local predicate, invoke a *decoding* verifier to do this.
3. In addition, select random $i \in I$ and ask decoding verifier to output this symbol. Check consistency vs. π_i.



- Query Complexity

  outer query complexity → inner query complexity + 1

Composition with dPCPs

- Question: Does this composition preserve robustness?

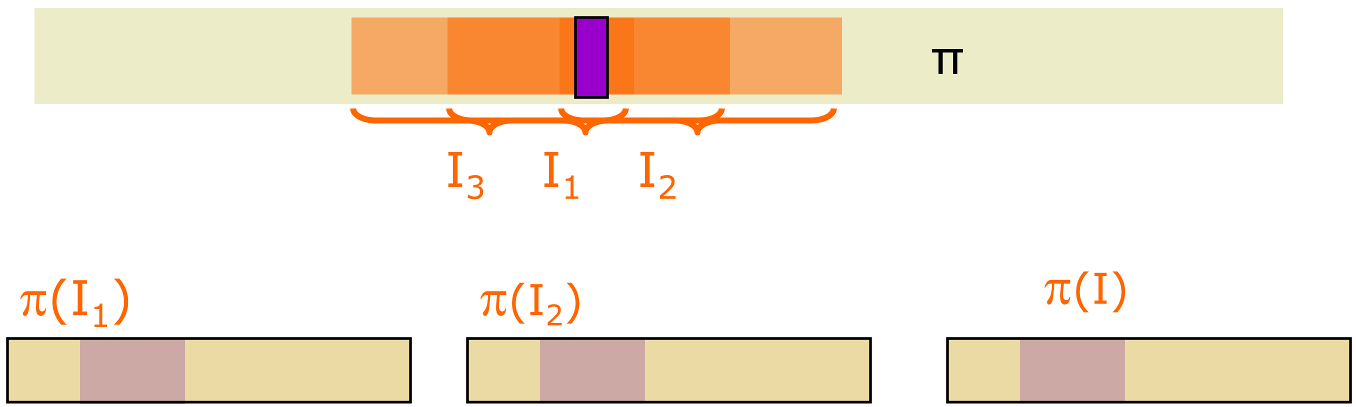
- Why do we care?
- If composed verifier is robust, by equivalence with Label Cover, we get a Label cover instance with small alphabet

- Alas, it is not robust! ...
  - Composed view consists of two parts – (view in outer and inner verifiers), each part can be completed to an accepting view. (robust soundness > ½)

- Easy to fix:

  Instead of decoding from inner proof and comparing to outer proof symbol, compare inner proofs *to each other*!

NEW: Robust Composition



<u>Composed verifier:</u>

1) select a random symbol i, consider all windows containing it.
2) Choose d such windows $I_1,\ldots I_d$
3) Run the inner verifier on each $\pi(I_j)$. Ask each to decode π_i
4) Accept iff all inner verifiers accept, and all answers are equal

Robust Composition

replace the consistency query

with decoding many local views simultaneously.

Outer query complexity → d * (inner query complexity)

Preserves robust soundness !!

4) Accept iff all inner verifiers accept, and all answers are equal

Robust Composition

- By equivalence with label-cover
  we get a label cover with smaller alphabet size.

- Perform repeatedly to obtain [MR'08] result
  - For every alphabet Σ and error $\delta = 1/\log|\Sigma|$, Gap(1, δ)-LC is NP-hard.

Open Question

- Error – Alphabet Relation
  - Parallel Repetition obtains $\delta = 1/poly|\Sigma|$
    - while
  - [MR'08] and new composition only yield $\delta = 1/\log|\Sigma|$

- Sliding Scale Conjecture
  - For every alphabet Σ and error $\delta = 1/poly|\Sigma|$, Gap(1, δ)-LC is NP-hard.

THANK YOU