BILL AND NATHAN, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!
Upper and Lower Bounds (PCP) on Approx For MAX3SAT
In this section we assume \( P \neq NP \).
In this section we assume $P \neq NP$.

If we say

**The Hokey Pokey cannot be approx better than BLAH**
In this section we assume $P \neq NP$.

If we say

The Hokey Pokey cannot be approx better than BLAH

We mean

If can approx The Hokey Pokey better than BLAH then $P = NP$
In this section we assume $P \neq NP$.

If we say

**The Hokey Pokey cannot be approx better than BLAH**

We mean

If can approx The Hokey Pokey better than BLAH then $P = NP$

If we say **Alg A** we mean **Poly time Alg A**.
In this section we assume $P \neq NP$.

If we say

**The Hokey Pokey cannot be approx better than BLAH**

We mean

If can approx The Hokey Pokey better than BLAH then $P = NP$

If we say **Alg A** we mean **Poly time Alg A**.

If we say **rand Alg A** we mean **Randomized Poly time Alg A**.
1. Input $\phi = C_1 \land \cdots \land C_m$, each $C_i$ is a $\lor$ of 3 literals.

2. Output The max number of clauses that can be satisfied.

Is there a $\delta < 1$ and an alg $A$ such that $A(\phi) \geq (1 - \delta) \text{MAX3SAT}(\phi)$?

Yes. Next Slide.
1. **Input** $\phi = C_1 \land \cdots \land C_m$, each $C_i$ is a $\lor$ of 3 literals.
MAX3SAT

1. **Input** $\phi = C_1 \land \cdots \land C_m$, each $C_i$ is a $\lor$ of 3 literals.
2. **Output** The max number of clauses that can be satisfied.
MAX3SAT

1. Input $\phi = C_1 \land \cdots \land C_m$, each $C_i$ is a $\lor$ of 3 literals.
2. Output The max number of clauses that can be satisfied.

Is there a $\delta < 1$ and an alg $A$ such that
MAX3SAT

1. **Input** $\phi = C_1 \land \cdots \land C_m$, each $C_i$ is a $\lor$ of 3 literals.
2. **Output** The max number of clauses that can be satisfied.

Is there a $\delta < 1$ and an alg $A$ such that

$$A(\phi) \geq (1 - \delta)\text{MAX3SAT}(\phi)$$
MAX3SAT

1. **Input** $\phi = C_1 \land \cdots \land C_m$, each $C_i$ is a $\lor$ of 3 literals.
2. **Output** The max number of clauses that can be satisfied.

Is there a $\delta < 1$ and an alg $A$ such that

$$A(\phi) \geq (1 - \delta)\text{MAX3SAT}(\phi)$$

Yes.
MAX3SAT

1. **Input** $\phi = C_1 \land \cdots \land C_m$, each $C_i$ is a $\lor$ of 3 literals.
2. **Output** The max number of clauses that can be satisfied.

Is there a $\delta < 1$ and an alg $A$ such that

$$A(\phi) \geq (1 - \delta)\text{MAX3SAT}(\phi)$$

Yes.

Next Slide
Approx for MAX3SAT

\textbf{Thm} (\exists) \textbf{rand alg} \ A \ st \ A(\phi) \geq \frac{7}{8} \text{MAX3SAT}(\phi).
Approx for MAX3SAT

**Thm** (∃) rand alg A st \( A(\phi) \geq \frac{7}{8} \text{MAX3SAT}(\phi) \).

1. Input \( \phi = C_1 \land \cdots \land C_m \).
Approx for MAX3SAT

**Thm** (∃) rand alg A st $A(\phi) \geq \frac{7}{8} \text{MAX3SAT}(\phi)$.  

1. Input $\phi = C_1 \land \cdots \land C_m$.  
2. Assign each var T or F at Random.
Approx for MAX3SAT

Thm \((\exists)\) rand alg A st \(A(\phi) \geq \frac{7}{8} \text{MAX3SAT}(\phi)\).

1. Input \(\phi = C_1 \land \cdots \land C_m\).
2. Assign each var T or F at Random.

Its just that easy!
Approx for MAX3SAT

**Thm (∃) rand alg A st A(φ) ≥ \( \frac{7}{8} \) MAX3SAT(φ).**

1. Input \( φ = C_1 \land \cdots \land C_m \).
2. Assign each var T or F at Random.

It's just that easy! Why does this work?
**Thm** (∃) rand alg A st A(φ) ≥ \( \frac{7}{8} \) MAX3SAT(φ).

1. Input \( \phi = C_1 \land \cdots \land C_m \).
2. Assign each var T or F at Random.

It’s just that easy! Why does this work?

Let \( C \) be a clause. The prob that \( C \) is satisfied is \( \frac{7}{8} \).
Approx for MAX3SAT

**Thm** (∃) rand alg A st $A(\phi) \geq \frac{7}{8} \text{MAX3SAT}(\phi)$.

1. Input $\phi = C_1 \land \cdots \land C_m$.
2. Assign each var T or F at Random.

Its just that easy! Why does this work?

Let $C$ be a clause. The prob that $C$ is satisfied is $\frac{7}{8}$.

By Lin of ExpV, expected number of $C_i$ satisfied is $\frac{7m}{8}$. 
Approx for MAX3SAT

**Thm** (∃) rand alg A st A(φ) ≥ \( \frac{7}{8} \) MAX3SAT(φ).

1. Input \( φ = C_1 \land \cdots \land C_m \).
2. Assign each var T or F at Random.

Its just that easy! Why does this work?

Let \( C \) be a clause. The prob that \( C \) is satisfied is \( \frac{7}{8} \).
By Lin of ExpV, expected number of \( C_i \) satisfied is \( \frac{7m}{8} \).
Note that MAX3SAT ≤ \( m \).
Approx for MAX3SAT

Thm \( \exists \) \text{ rand alg} \ A \ st \ A(\phi) \geq \frac{7}{8} \text{MAX3SAT}(\phi).

1. Input \( \phi = C_1 \land \cdots \land C_m \).
2. Assign each var T or F at Random.

Its just that easy! Why does this work?

Let \( C \) be a clause. The prob that \( C \) is satisfied is \( \frac{7}{8} \).

By Lin of ExpV, expected number of \( C_i \) satisfied is \( \frac{7m}{8} \).

Note that \( \text{MAX3SAT} \leq m \).

Hence \( A(\phi) \geq \frac{7}{8} \text{MAX3SAT}(\phi) \).
Thm (∃) rand alg A st A(φ) ≥ \( \frac{7}{8} \) MAX3SAT(φ).

1. Input \( φ = C_1 \land \cdots \land C_m \).
2. Assign each var T or F at Random.

Its just that easy! Why does this work?

Let C be a clause. The prob that C is satisfied is \( \frac{7}{8} \).

By Lin of ExpV, expected number of \( C_i \) satisfied is \( \frac{7m}{8} \).

Note that MAX3SAT \( \leq m \).

Hence \( A(φ) \geq \frac{7}{8} \) MAX3SAT(φ).

Note This rand alg can be made det by method of cond prob.
Approx for Variants of MAX3SAT

1. If (\(\forall i\)) \(|C_i| = 3\) then have easy rand alg returns \(\geq \frac{7}{8}\) MAX3SAT(\(\phi\)).

2. If (\(\forall i\)) \(|C_i| = 3\) then have medium det alg returns \(\geq \frac{7}{8}\) MAX3SAT(\(\phi\)).

3. If (\(\forall i\)) \(|C_i| \leq 3\) then have easy rand alg returns \(\geq \frac{1}{2}\) MAX3SAT(\(\phi\)).

4. If (\(\forall i\)) \(|C_i| \leq 3\) then have medium det alg returns \(\geq \frac{1}{2}\) MAX3SAT(\(\phi\)).

5. If (\(\forall i\)) \(|C_i| \leq 3\) then have hard rand alg returns \(\geq \frac{7}{8}\) MAX3SAT(\(\phi\)).

People tried to get an app-alg to return \(\geq (\frac{7}{8} + \epsilon)\) MAX3SAT(\(\phi\)). Did they succeed? No. Now What?
Approx for Variants of MAX3SAT

1. If $(\forall i)[|C_i| = 3]$ then have easy rand alg returns $\geq \frac{7}{8} \text{MAX3SAT}(\phi)$. 
1. If \((\forall i)[|C_i| = 3]\) then have easy rand alg returns 
\[\geq \frac{7}{8} \text{MAX3SAT}(\phi)\].

2. If \((\forall i)[|C_i| = 3]\) then have medium det alg returns 
\[\geq \frac{7}{8} \text{MAX3SAT}(\phi)\].
Approx for Variants of MAX3SAT

1. If $(\forall i)[|C_i| = 3]$ then have easy rand alg returns $\geq \frac{7}{8}\text{MAX3SAT}(\phi)$.

2. If $(\forall i)[|C_i| = 3]$ then have medium det alg returns $\geq \frac{7}{8}\text{MAX3SAT}(\phi)$.

3. If $(\forall i)[|C_i| \leq 3]$ then have easy rand alg returns $\geq \frac{1}{2}\text{MAX3SAT}(\phi)$.
Approx for Variants of MAX3SAT

1. If $(\forall i)[|C_i| = 3]$ then have **easy** rand alg returns $\geq \frac{7}{8} \text{MAX3SAT}(\phi)$.

2. If $(\forall i)[|C_i| = 3]$ then have **medium** det alg returns $\geq \frac{7}{8} \text{MAX3SAT}(\phi)$.

3. If $(\forall i)[|C_i| \leq 3]$ then have **easy** rand alg returns $\geq \frac{1}{2} \text{MAX3SAT}(\phi)$.

4. If $(\forall i)[|C_i| \leq 3]$ then have **medium** det alg returns $\geq \frac{1}{2} \text{MAX3SAT}(\phi)$.

People tried to get an app-alg to return $\geq (\frac{7}{8} + \epsilon) \text{MAX3SAT}(\phi)$.

Did they succeed?

No.

Now What?
Approx for Variants of MAX3SAT

1. If $(\forall i)[|C_i| = 3]$ then have easy rand alg returns $\geq \frac{7}{8}\text{MAX3SAT}(\phi)$.

2. If $(\forall i)[|C_i| = 3]$ then have medium det alg returns $\geq \frac{7}{8}\text{MAX3SAT}(\phi)$.

3. If $(\forall i)[|C_i| \leq 3]$ then have easy rand alg returns $\geq \frac{1}{2}\text{MAX3SAT}(\phi)$.

4. If $(\forall i)[|C_i| \leq 3]$ then have medium det alg returns $\geq \frac{1}{2}\text{MAX3SAT}(\phi)$.

5. If $(\forall i)[|C_i| \leq 3]$ then have hard rand alg returns $\geq \frac{7}{8}\text{MAX3SAT}(\phi)$.
Approx for Variants of MAX3SAT

1. If $(\forall i)[|C_i| = 3]$ then have easy rand alg returns $\geq \frac{7}{8} \text{MAX3SAT}(\phi)$.

2. If $(\forall i)[|C_i| = 3]$ then have medium det alg returns $\geq \frac{7}{8} \text{MAX3SAT}(\phi)$.

3. If $(\forall i)[|C_i| \leq 3]$ then have easy rand alg returns $\geq \frac{1}{2} \text{MAX3SAT}(\phi)$.

4. If $(\forall i)[|C_i| \leq 3]$ then have medium det alg returns $\geq \frac{1}{2} \text{MAX3SAT}(\phi)$.

5. If $(\forall i)[|C_i| \leq 3]$ then have hard rand alg returns $\geq \frac{7}{8} \text{MAX3SAT}(\phi)$.

People tried to get an app-alg to return $\geq (\frac{7}{8} + \epsilon) \text{MAX3SAT}(\phi)$. 
Approx for Variants of MAX3SAT

1. If \((\forall i)[|C_i| = 3]\) then have easy rand alg returns \(\geq \frac{7}{8}\text{MAX3SAT}(\phi)\).

2. If \((\forall i)[|C_i| = 3]\) then have medium det alg returns \(\geq \frac{7}{8}\text{MAX3SAT}(\phi)\).

3. If \((\forall i)[|C_i| \leq 3]\) then have easy rand alg returns \(\geq \frac{1}{2}\text{MAX3SAT}(\phi)\).

4. If \((\forall i)[|C_i| \leq 3]\) then have medium det alg returns \(\geq \frac{1}{2}\text{MAX3SAT}(\phi)\).

5. If \((\forall i)[|C_i| \leq 3]\) then have hard rand alg returns \(\geq \frac{7}{8}\text{MAX3SAT}(\phi)\).

People tried to get an app-alg to return \(\geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi)\). Did they succeed?
Approx for Variants of MAX3SAT

1. If $(\forall i)[|C_i| = 3]$ then have **easy** rand alg returns $\geq \frac{7}{8}\text{MAX3SAT}(\phi)$.

2. If $(\forall i)[|C_i| = 3]$ then have **medium** det alg returns $\geq \frac{7}{8}\text{MAX3SAT}(\phi)$.

3. If $(\forall i)[|C_i| \leq 3]$ then have **easy** rand alg returns $\geq \frac{1}{2}\text{MAX3SAT}(\phi)$.

4. If $(\forall i)[|C_i| \leq 3]$ then have **medium** det alg returns $\geq \frac{1}{2}\text{MAX3SAT}(\phi)$.

5. If $(\forall i)[|C_i| \leq 3]$ then have **hard** rand alg returns $\geq \frac{7}{8}\text{MAX3SAT}(\phi)$.

People tried to get an app-alg to return $\geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi)$. Did they succeed? No.
Approx for Variants of MAX3SAT

1. If \((\forall i)[|C_i| = 3]\) then have easy rand alg returns 
   \[\geq \frac{7}{8} \text{MAX3SAT}(\phi).\]

2. If \((\forall i)[|C_i| = 3]\) then have medium det alg returns 
   \[\geq \frac{7}{8} \text{MAX3SAT}(\phi).\]

3. If \((\forall i)[|C_i| \leq 3]\) then have easy rand alg returns 
   \[\geq \frac{1}{2} \text{MAX3SAT}(\phi).\]

4. If \((\forall i)[|C_i| \leq 3]\) then have medium det alg returns 
   \[\geq \frac{1}{2} \text{MAX3SAT}(\phi).\]

5. If \((\forall i)[|C_i| \leq 3]\) then have hard rand alg returns 
   \[\geq \frac{7}{8} \text{MAX3SAT}(\phi).\]

People tried to get an app-alg to return \[\geq (\frac{7}{8} + \epsilon) \text{MAX3SAT}(\phi).\]
Did they succeed? No. Now What?
There is a Limit To How Well You Can Approx

We will show there is some $\delta < \frac{1}{8}$ such that there is NO app-alg that returns

$$\geq (1 - \delta)\text{MAX3SAT}(\phi).$$
There is a Limit To How Well You Can Approx

We will show there is some $\delta < \frac{1}{8}$ such that there is NO app-alg that returns

$$\geq (1 - \delta)\text{MAX3SAT}(\phi).$$

Hence cannot keep getting better and better approx.
There is a Limit To How Well You Can Approx

We will show there is some $\delta < \frac{1}{8}$ such that there is NO app-alg that returns

$$\geq (1 - \delta)\text{MAX3SAT}(\phi).$$

Hence cannot keep getting better and better approx.

**Consequence** $\exists \epsilon < \frac{1}{8}$, $\neg \exists$ alg $A$, $A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi)$. 
There is a Limit To How Well You Can Approx

We will show there is some $\delta < \frac{1}{8}$ such that there is NO app-alg that returns

$$\geq (1 - \delta)\text{MAX3SAT}(\phi).$$

Hence cannot keep getting better and better approx.

**Consequence** $\exists \epsilon < \frac{1}{8}$, $\neg \exists$ alg $A$, $A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi)$. The value of $\epsilon$ is buried in the machinery of PCP though it could be determined.
There is a Limit To How Well You Can Approx

We will show there is some $\delta < \frac{1}{8}$ such that there is NO app-alg that returns

$$\geq (1 - \delta)\text{MAX3SAT}(\phi).$$

Hence cannot keep getting better and better approx.

**Consequence** $\exists \epsilon < \frac{1}{8}$, $\neg \exists$ alg $A$, $A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi)$.

The value of $\epsilon$ is buried in the machinery of PCP though it could be determined.

Likely end up with something like:
There is a Limit To How Well You Can Approx

We will show there is some $\delta < \frac{1}{8}$ such that there is NO app-alg that returns

$$\geq (1 - \delta) \text{MAX3SAT}(\phi).$$

Hence cannot keep getting better and better approx.

**Consequence** $\exists \epsilon < \frac{1}{8}$, $\neg \exists$ alg $A$, $A(\phi) \geq (\frac{7}{8} + \epsilon) \text{MAX3SAT}(\phi)$.

The value of $\epsilon$ is buried in the machinery of PCP though it could be determined.

Likely end up with something like: There is no Alg $A$ such that
There is a Limit To How Well You Can Approx

We will show there is some $\delta < \frac{1}{8}$ such that there is NO app-alg that returns

$$\geq (1 - \delta)\text{MAX3SAT}(\phi).$$

Hence cannot keep getting better and better approx.

Consequence $\exists \epsilon < \frac{1}{8}$, $\neg \exists$ alg $A$, $A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi)$.

The value of $\epsilon$ is buried in the machinery of PCP though it could be determined.

 Likely end up with something like:
There is no Alg $A$ such that

$$A(\phi) \geq \frac{10^{40} - 1}{10^{40}}\text{MAX3SAT}(\phi).$$
There is a Limit To How Well You Can Approx

We will show there is some $\delta < \frac{1}{8}$ such that there is NO app-alg that returns

$$\geq (1 - \delta)\text{MAX3SAT}(\phi).$$

Hence cannot keep getting better and better approx.

**Consequence** $\exists \epsilon < \frac{1}{8}, \neg \exists$ alg $A$, $A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi)$.

The value of $\epsilon$ is buried in the machinery of PCP though it could be determined.

Likely end up with something like:
There is no Alg $A$ such that

$$A(\phi) \geq \frac{10^{40} - 1}{10^{40}}\text{MAX3SAT}(\phi).$$

(An alg that does better and better is a Poly Time Approx Scheme (PTAS). We show there is no PTAS for MAX3SAT.)
Thm
\( \forall \epsilon > 0, \neg \exists \text{alg } A, A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi). \)
Better Lower Bounds Are Known

**Thm**

∀ ϵ > 0, ¬∃ alg A, A(φ) ≥ (\(\frac{7}{8}\) + ϵ)MAX3SAT(φ).

So can’t even do a wee bit better,
Better Lower Bounds Are Known

\textbf{Thm}
\[ \forall \epsilon > 0, \neg \exists \text{ alg } A, A(\phi) \geq (\frac{7}{8} + \epsilon) \text{MAX3SAT}(\phi). \]
So can’t even do a wee bit better,
If Erika says she has an alg that returns \[ \geq (\frac{7}{8} + \frac{1}{10^{40}}) \text{MAX3SAT}(\phi) \]
then either

\begin{enumerate}
\item The rand and poly app-algs that got \frac{7}{8} \text{MAX3SAT}(\phi) are easy.
\item The lower bound uses PCP machinery.
\item The alg and the lower bounds have nothing to do with each other and yet yield matching upper and lower bounds at \frac{7}{8}.
\end{enumerate}
Better Lower Bounds Are Known

**Thm**
\[ \forall \epsilon > 0, \neg \exists \text{ alg } A, A(\phi) \geq (\frac{7}{8} + \epsilon) \text{MAX3SAT}(\phi). \]
So can’t even do a wee bit better,

If Erika says she has an alg that returns \[ \geq (\frac{7}{8} + \frac{1}{1040}) \text{MAX3SAT}(\phi) \]
then either (a) Erika has proven P = NP or
Thm
\[ \forall \epsilon > 0, \neg \exists \text{ alg } A, A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi). \]
So can’t even do a wee bit better,

If Erika says she has an alg that returns \[ \geq (\frac{7}{8} + \frac{1}{1040})\text{MAX3SAT}(\phi) \]
then either (a) Erika has proven \( P = NP \) or (b) Erika is mistaken.
Better Lower Bounds Are Known

**Thm**

\[ \forall \epsilon > 0, \neg \exists \text{alg } A, A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi). \]

So can’t even do a wee bit better,

If Erika says she has an alg that returns \( \geq (\frac{7}{8} + \frac{1}{10^{40}})\text{MAX3SAT}(\phi) \)
then either (a) Erika has proven \( P = NP \) or (b) Erika is mistaken.

Yet another example of the explanatory power of \( P \neq NP \)
**Better Lower Bounds Are Known**

**Thm**
\[ \forall \epsilon > 0, \neg \exists \text{ alg } A, A(\phi) \geq (\frac{7}{8} + \epsilon) \text{MAX3SAT}(\phi). \]
So can’t even do a wee bit better,

If Erika says she has an alg that returns \[ \geq (\frac{7}{8} + \frac{1}{10^{40}}) \text{MAX3SAT}(\phi) \]
then either (a) Erika has proven P = NP or (b) Erika is mistaken.

**Yet another example of the explanatory power of P \neq NP**

Note that
Better Lower Bounds Are Known

Thm
\[ \forall \epsilon > 0, \neg \exists \text{alg } A, A(\phi) \geq \left( \frac{7}{8} + \epsilon \right) \text{MAX3SAT}(\phi). \]
So can’t even do a wee bit better,
If Erika says she has an alg that returns \( \geq \left( \frac{7}{8} + \frac{1}{10^40} \right) \text{MAX3SAT}(\phi) \) then either (a) Erika has proven \( P = NP \) or (b) Erika is mistaken.

Yet another example of the explanatory power of \( P \neq NP \)

Note that
1. The rand and poly app-algs that got \( \frac{7}{8} \text{MAX3SAT}(\phi) \) are easy.
Thm
∀ε > 0, ¬∃ alg A, A(φ) ≥ (7/8 + ε)MAX3SAT(φ).

So can’t even do a wee bit better,

If Erika says she has an alg that returns ≥ (7/8 + 1/10^40)MAX3SAT(φ) then either (a) Erika has proven P = NP or (b) Erika is mistaken.

Yet another example of the explanatory power of P ≠ NP

Note that

1. The rand and poly app-algs that got 7/8 MAX3SAT(φ) are easy.
2. The lower bound uses PCP machinery.
Better Lower Bounds Are Known

\textbf{Thm}
\[ \forall \epsilon > 0, \neg \exists \text{alg } A, A(\phi) \geq (\frac{7}{8} + \epsilon)\text{MAX3SAT}(\phi). \]
So can’t even do a wee bit better,
If Erika says she has an alg that returns \( \geq (\frac{7}{8} + \frac{1}{1040})\text{MAX3SAT}(\phi) \) then either (a) Erika has proven \( P = NP \) or (b) Erika is mistaken.

\textbf{Yet another example of the explanatory power of} \( P \neq NP \)

Note that
1. The rand and poly app-algs that got \( \frac{7}{8} \text{MAX3SAT}(\phi) \) are easy.
2. The lower bound uses PCP machinery.
3. The alg and the lower bounds have \textbf{nothing to do with each other} and yet yield \textbf{matching} upper and lower bounds at \( \frac{7}{8} \).
Recall Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$ such that $A \in \text{PCP}(q, d \lg n, \epsilon)$. 
**Recall** Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$ such that $A \in \text{PCP}(q, d \lg n, \epsilon)$.

Let $x \in \{0, 1\}^n$. This is the input to the PCP.
Recall Let \( A \in \text{NP} \) and \( \epsilon > 0 \). Then \( \exists q, d \in \mathbb{N} \) such that \( A \in \text{PCP}(q, d \lg n, \epsilon) \).
Let \( x \in \{0, 1\}^n \). This is the input to the PCP.
We form a Boolean formula as follows.
Recall Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$ such that $A \in \text{PCP}(q, d \lfloor \log n \rfloor, \epsilon)$. Let $x \in \{0, 1\}^n$. This is the input to the PCP. We form a Boolean formula as follows.

**The Vars** For every $\tau\sigma \in \{0, 1\}^{d \lfloor \log n \rfloor + q}$ one can run the PCP with random string $\tau$ and bit-answers $\sigma$. From these simulations you can find all possible bit-queries. There are $\leq 2^{d \lfloor \log n \rfloor + q} = 2^q n^d$ bit queries. These will be variables.
Recall Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$ such that $A \in \text{PCP}(q, d \lg n, \epsilon)$.

Let $x \in \{0,1\}^n$. This is the input to the PCP.

We form a Boolean formula as follows.

The Vars For every $\tau \sigma \in \{0,1\}^{d \lg n + q}$ one can run the PCP with random string $\tau$ and bit-answers $\sigma$. From these simulations you can find all possible bit-queries. There are $\leq 2^{d \lg n + q} = 2^q n^d$ bit queries. These will be variables.

Parts of the Formula For every $\tau \in \{0,1\}^{d \lg n}$ we form $\psi_\tau$. 
Recall Let \( A \in \text{NP} \) and \( \epsilon > 0 \). Then \( \exists q, d \in \mathbb{N} \) such that \( A \in \text{PCP}(q, d \lg n, \epsilon) \).

Let \( x \in \{0, 1\}^n \). This is the input to the PCP.

We form a Boolean formula as follows.

The Vars For every \( \tau\sigma \in \{0, 1\}^{d \lg n + q} \) one can run the PCP with random string \( \tau \) and bit-answers \( \sigma \). From these simulations you can find all possible bit-queries. There are \( \leq 2^{d \lg n + q} = 2^q n^d \) bit queries. These will be variables.

Parts of the Formula For every \( \tau \in \{0, 1\}^{d \lg n} \) we form \( \psi_\tau \).

Use \( \tau \) as the random string. Simulate all possible query paths to find the relevant vars.
Recall Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$ such that $A \in \text{PCP}(q, d \lg n, \epsilon)$.

Let $x \in \{0, 1\}^n$. This is the input to the PCP.

We form a Boolean formula as follows.

The Vars For every $\tau \sigma \in \{0, 1\}^{d \lg n + q}$ one can run the PCP with random string $\tau$ and bit-answers $\sigma$. From these simulations you can find all possible bit-queries. There are $\leq 2^{d \lg n + q} = 2^q n^d$ bit queries. These will be variables.

Parts of the Formula For every $\tau \in \{0, 1\}^{d \lg n}$ we form $\psi_\tau$.

Use $\tau$ as the random string. Simulate all possible query paths to find the relevant vars.

$\psi_\tau$ is the formula on those vars that is TRUE exactly when that setting of the variables makes this path accept.
A Very Small Example

Imagine the following.

$$\psi_{1101} = (q_{17} \land q_{84}) \lor (\neg q_{17} \land \neg q_{5}).$$
Imagine the following.
Using $\tau = 1101$ the PCP will query bit 17.
Imagine the following.
Using $\tau = 1101$ the PCP will query bit 17.
If bit 17 is 1 then query bit 84. If bit 17 is 0 then query bit 5.

$\psi_{1101} = (q_{17} \land q_{84}) \lor (\neg q_{17} \land \neg q_{5})$. 
Imagine the following. Using $\tau = 1101$ the PCP will query bit 17. If bit 17 is 1 then query bit 84. If bit 17 is 0 then query bit 5. If bit 17 is 1 and then bit 84 is 1 then accept. If bit 17 is 0 and then bit 5 is 0 then accept. All else reject.
Imagine the following.

Using $τ = 1101$ the PCP will query bit 17.

If bit 17 is 1 then query bit 84. If bit 17 is 0 then query bit 5.

If bit 17 is 1 and then bit 84 is 1 then accept.
If bit 17 is 0 and then bit 5 is 0 then accept.
All else reject.

$ψ_{1101} = (q_{17} \land q_{84}) \lor (\neg q_{17} \land \neg q_{5})$. 
Max Number of Clauses

In general case we will turn $\psi_\tau$ into a 3CNF.
Max Number of Clauses

In general case we will turn $\psi_\tau$ into a $3$CNF.
We do not have any control over how many clauses $\psi_\tau$ will have.
Max Number of Clauses

In general case we will turn $\psi_\tau$ into a 3CNF.
We do not have any control over how many clauses $\psi_\tau$ will have. But we do know that it uses $\leq 2^q$ variables.
Max Number of Clauses

In general case we will turn $\psi_\tau$ into a 3CNF. We do not have any control over how many clauses $\psi_\tau$ will have. But we do know that it uses $\leq 2^q$ variables.

**Def** $C(q)$ is max numb of clauses a 3CNF fml on $2^q$ vars has.
Max Number of Clauses

In general case we will turn $\psi_\tau$ into a 3CNF. We do not have any control over how many clauses $\psi_\tau$ will have. But we do know that it uses $\leq 2^q$ variables.

**Def** $C(q)$ is max numb of clauses a 3CNF fml on $2^q$ vars has.

**Note** Since $q$ is a constant, $C(q)$ is a constant.
Max Number of Clauses

In general case we will turn $\psi_\tau$ into a 3CNF. We do not have any control over how many clauses $\psi_\tau$ will have. But we do know that it uses $\leq 2^q$ variables.

**Def** $C(q)$ is max numb of clauses a 3CNF fml on $2^q$ vars has.

**Note** Since $q$ is a constant, $C(q)$ is a constant. We will use $C(q)$ later.
Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $x \in \{0, 1\}^n$. This is the input to the PCP.
Final Formula

Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $x \in \{0, 1\}^n$. This is the input to the PCP.
We have said how to take $\tau \in \{0, 1\}^{d \lg n}$ and form $\psi_\tau$. 
Final Formula

Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $x \in \{0, 1\}^n$. This is the input to the PCP.
We have said how to take $\tau \in \{0, 1\}^{d \lg n}$ and form $\psi_\tau$.

1. $\psi_\tau$ is on $\leq 2^q$ vars, a constant. Rewrite $\psi_\tau$ as a 3CNF.
Final Formula

Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$

Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.

Let $x \in \{0, 1\}^n$. This is the input to the PCP.

We have said how to take $\tau \in \{0, 1\}^{d \lg n}$ and form $\psi_\tau$.

1. $\psi_\tau$ is on $\leq 2^q$ vars, a constant. Rewrite $\psi_\tau$ as a 3CNF.
2. $\psi_\tau$ has $\leq C(q)$ clauses. Add clauses of the form $(x \lor x \lor x)$ with new vars $x$ to get exactly $C(q)$ clauses.
Final Formula

Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $x \in \{0, 1\}^n$. This is the input to the PCP.
We have said how to take $\tau \in \{0, 1\}^{d \lg n}$ and form $\psi_\tau$.

1. $\psi_\tau$ is on $\leq 2^q$ vars, a constant. Rewrite $\psi_\tau$ as a 3CNF.
2. $\psi_\tau$ has $\leq C(q)$ clauses. Add clauses of the form $(x \lor x \lor x)$ with new vars $x$ to get exactly $C(q)$ clauses.
3. Let $\psi_x$ be the $\land$ of all the $\psi_\tau$. 
Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $x \in \{0, 1\}^n$. This is the input to the PCP.
We have said how to take $\tau \in \{0, 1\}^{d \lg n}$ and form $\psi_\tau$.

1. $\psi_\tau$ is on $\leq 2^q$ vars, a constant. Rewrite $\psi_\tau$ as a 3CNF.
2. $\psi_\tau$ has $\leq C(q)$ clauses. Add clauses of the form $(x \lor x \lor x)$
   with new vars $x$ to get exactly $C(q)$ clauses.
3. Let $\psi_x$ be the $\land$ of all the $\psi_\tau$.
4. Note that $\psi_x$ is 3CNF.
Final Formula

Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $x \in \{0, 1\}^n$. This is the input to the PCP.
We have said how to take $\tau \in \{0, 1\}^{d \lg n}$ and form $\psi_\tau$.

1. $\psi_\tau$ is on $\leq 2^q$ vars, a constant. Rewrite $\psi_\tau$ as a 3CNF.
2. $\psi_\tau$ has $\leq C(q)$ clauses. Add clauses of the form $(x \lor x \lor x)$
   with new vars $x$ to get exactly $C(q)$ clauses.
3. Let $\psi_x$ be the $\land$ of all the $\psi_\tau$.
4. Note that $\psi_x$ is 3CNF.
5. $\psi_x$ has $2^{d \lg n}C(q) = n^dC(q)$ clauses.
Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $x \in \{0, 1\}^n$. This is the input to the PCP.
We have said how to take $\tau \in \{0, 1\}^{d \lg n}$ and form $\psi_\tau$.

1. $\psi_\tau$ is on $\leq 2^q$ vars, a constant. Rewrite $\psi_\tau$ as a 3CNF.
2. $\psi_\tau$ has $\leq C(q)$ clauses. Add clauses of the form $(x \lor x \lor x)$ with new vars $x$ to get exactly $C(q)$ clauses.
3. Let $\psi_x$ be the $\land$ of all the $\psi_\tau$.
4. Note that $\psi_x$ is 3CNF.
5. $\psi_x$ has $2^{d \lg n}C(q) = n^d C(q)$ clauses.
6. Note that $\psi_x$ is in 3CNF Form and has $C(q)n^d$ clauses.
Let $A \in \text{NP}$ and $\epsilon > 0$. Then $\exists q, d \in \mathbb{N}$
Let $A \in \text{PCP}(q, d \lg n, \epsilon)$.
Let $x \in \{0, 1\}^n$. This is the input to the PCP.
We have said how to take $\tau \in \{0, 1\}^{d \lg n}$ and form $\psi_\tau$.

1. $\psi_\tau$ is on $\leq 2^q$ vars, a constant. Rewrite $\psi_\tau$ as a 3CNF.
2. $\psi_\tau$ has $\leq C(q)$ clauses. Add clauses of the form $(x \lor x \lor x)$ with new vars $x$ to get exactly $C(q)$ clauses.
3. Let $\psi_x$ be the $\bigwedge$ of all the $\psi_\tau$.
4. Note that $\psi_x$ is 3CNF.
5. $\psi_x$ has $2^{d \lg n}C(q) = n^d C(q)$ clauses.
6. Note that $\psi_x$ is in 3CNF Form and has $C(q)n^d$ clauses.

Going from $x$ to $\psi_x$ takes time poly in $|x| = n$. 

Assume BWOC \((\forall \delta < 1)\) \(\text{MAX3SAT}\) is \((1 - \delta)\)-approximable.
Assume BWOC ($\forall \delta < 1$) MAX3SAT is $(1 - \delta)$-approximable. We pick $\delta$ later. It will matter.
MAX3SAT is Not PTAS: Set Up

Assume BWOC ($\forall \delta < 1$) MAX3SAT is $(1 - \delta)$-approximable. We pick $\delta$ later. It will matter.
We call the approx algorithm that achieves this app-alg.
MAX3SAT is Not PTAS: Set Up

Assume BWOC ($\forall \delta < 1$) MAX3SAT is $(1 - \delta)$-approximable. We pick $\delta$ later. It will matter.
We call the approx algorithm that achieves this app-alg.

Let $A \in \text{NP}$. We pick $\varepsilon$ later. It won’t matter.
MAX3SAT is Not PTAS: Set Up

Assume BWOC ($\forall \delta < 1$) MAX3SAT is $(1 - \delta)$-approximable. We pick $\delta$ later. It will matter.

We call the approx algorithm that achieves this app-alg.

Let $A \in \text{NP}$. We pick $\epsilon$ later. It won’t matter.

By PCP Thm ($\exists d, q \in \mathbb{N}) [A \in \text{PCP}(q, d \log n, \epsilon)]$.
MAX3SAT is Not PTAS: Set Up

Assume BWOC ($\forall \delta < 1$) MAX3SAT is $(1 - \delta)$-approximable. We pick $\delta$ later. It will matter.

We call the approx algorithm that achieves this **app-alg**.

Let $A \in \text{NP}$. We pick $\epsilon$ later. It won’t matter.

**By PCP Thm** ($\exists d, q \in \mathbb{N})[A \in \text{PCP}(q, d \log n, \epsilon)]$.

If we run the PCP with oracle $y$ we say **PCP$^y$**.
MAX3SAT is Not PTAS: Set Up

Assume BWOC ($\forall \delta < 1$) MAX3SAT is $(1 - \delta)$-approximable.
We pick $\delta$ later. It will matter.
We call the approx algorithm that achieves this $\text{app-alg}$.

Let $A \in \text{NP}$. We pick $\epsilon$ later. It won’t matter.

**By PCP Thm** ($\exists d, q \in \mathbb{N})[A \in \text{PCP}(q, d \log n, \epsilon)]$.

If we run the PCP with oracle $y$ we say $\text{PCP}^y$.

We use $\text{app-alg}$ and the PCP to obtain $A \in \text{P}$.
MAX3SAT is Not PTAS: A in P Algorithm

1. Input $x$.
2. Form the 3CNF formula $\psi_x$.
3. Apply the approx to $\psi_x$.
4. We will pick $\epsilon, \delta$ such that there is a gap between what the approx yields if $x \in A$ and if $x \not\in A$. Details on next "few" slides.

We will then finish the algorithm.
MAX3SAT is Not PTAS: A in P Algorithm

1. Input $x$. 
1. Input $x$.
2. Form the 3CNF formula $\psi_x$. 
MAX3SAT is Not PTAS: A in P Algorithm

1. Input $x$.
2. Form the 3CNF formula $\psi_x$.
3. Apply the approx to $\psi_x$. 
1. Input $x$.
2. Form the 3CNF formula $\psi_x$.
3. Apply the approx to $\psi_x$.
4. We will pick $\epsilon, \delta$ such that there is a gap between what the approx yields if $x \in A$ and if $x \notin A$. Details on next “few” slides.
MAX3SAT is Not PTAS: A in P Algorithm

1. Input $x$.
2. Form the 3CNF formula $\psi_x$.
3. Apply the approx to $\psi_x$.
4. We will pick $\epsilon, \delta$ such that there is a gap between what the approx yields if $x \in A$ and if $x \notin A$. Details on next “few” slides.

We will then finish the algorithm.
MAX3SAT is Not PTAS: $x \in A$ Case

Assume $x \in A$. 
Assume $x \in A$.

Then there is an oracle $y$ so that, for all $\tau$, the PCP, with $\tau$, and using $y$ for answers, accepts.
MAX3SAT is Not PTAS: \( x \in A \) Case

Assume \( x \in A \).

Then there is an oracle \( y \) so that, for all \( \tau \), the PCP, with \( \tau \), and using \( y \) for answers, accepts.

Formally

\[
(\exists y)(\forall \tau \in \{0, 1\}^{d\lg n})[\text{PCP}^y(x, \tau) \text{ ACCEPTS}].
\]
Assume $x \in A$.
Then there is an oracle $y$ so that, for all $\tau$, the PCP, with $\tau$, and using $y$ for answers, accepts.
Formally

$$(\exists y)(\forall \tau \in \{0,1\}^{d \lg n})[\text{PCP}^y(x, \tau) \text{ ACCEPTS}].$$

Hence there is a way to satisfy all $n^d C(q)$ clauses of $\psi_\tau$ simul. 
So $\text{OPT}(\psi_x) = n^d C(q)$. 
Assume $x \notin A$. 
MAX3SAT is Not PTAS: $x \notin A$ Case

Assume $x \notin A$.

For all oracles $y$, for at most $\epsilon$ of the $\tau$, the PCP, with $\tau$, and using $y$ for answers, accepts.
MAX3SAT is Not PTAS: $x \notin A$ Case

Assume $x \notin A$.

For all oracles $y$, for at most $\epsilon$ of the $\tau$, the PCP, with $\tau$, and using $y$ for answers, accepts.

Formally
MAX3SAT is Not PTAS: $x \notin A$ Case

Assume $x \notin A$.

For all oracles $y$, for at most $\epsilon$ of the $\tau$, the PCP, with $\tau$, and using $y$ for answers, accepts.

Formally

$(\forall y \in \{0, 1\}^q)$

For $\leq \epsilon(2^{d\log n})$ of the $\tau \in \{0, 1\}^{d\log n}$[PCP$^y(x, \tau)$ACCEPTS].
If $x \notin A$ How Many Clauses Satisfied?

Let $y$ be the oracle (Truth Assignment) that yields $\text{OPT}(\psi_x)$
If \( x \notin A \) How Many Clauses Satisfied?

Let \( y \) be the oracle (Truth Assignment) that yields \( \text{OPT}(\psi_x) \)

\[
\psi_x = \bigwedge \psi_{\tau}
\]

Recall Each \( \psi_x \) has exactly \( C(q) \) clauses.

At most \( \epsilon \) of the \( \tau \)'s are satisfied.

Worst case For \( \phi_{\tau} / \in \text{SAT} \), \( \text{OPT}(\phi_{\tau}) = C(q) - 1 \).

So Number of clauses satisfied is

\[
\text{OPT}(\psi_x) = n d C(q) + (1 - \epsilon) n d (C(q) - 1) = n d (C(q) - 1) + \epsilon C(q) \]
If $x \notin A$ How Many Clauses Satisfied?

Let $y$ be the oracle (Truth Assignment) that yields $\text{OPT}(\psi_x)$

$$\psi_x = \bigwedge \psi_T$$

**Recall** Each $\psi_x$ has exactly $C(q)$ clauses.
If \( x \notin A \) How Many Clauses Satisfied?

Let \( y \) be the oracle (Truth Assignment) that yields \( \text{OPT}(\psi_x) \)

\[
\psi_x = \bigwedge \psi_{\tau}
\]

Recall Each \( \psi_x \) has exactly \( C(q) \) clauses.
At most \( \epsilon \) of the \( \tau \)'s are satisfied.
Worst case For \( \phi_{\tau} \notin \text{SAT} \), \( \text{OPT}(\phi_{\tau}) = C(q) - 1 \).
If \( x \notin A \) How Many Clauses Satisfied?

Let \( y \) be the oracle (Truth Assignment) that yields \( \text{OPT}(\psi_x) \)

\[
\psi_x = \bigwedge \psi_\tau
\]

Recall Each \( \psi_x \) has exactly \( C(q) \) clauses. At most \( \epsilon \) of the \( \tau \)'s are satisfied.

Worst case For \( \phi_\tau \notin \text{SAT} \), \( \text{OPT}(\phi_\tau) = C(q) - 1 \).

So Number of clauses satisfied is

\[
\epsilon n^d C(q) + (1 - \epsilon)n^d(C(q) - 1) = n^d(\epsilon C(q) + (1 - \epsilon)(C(q) - 1))
\]

\[
= n^d(\epsilon C(q) + C(q) - \epsilon C(q) - 1 + \epsilon) = n^d(C(q) - 1 + \epsilon)
\]
Apply Approx and See What Happens
Apply Approx and See What Happens

\[ x \in A \text{ MAX3SAT}(\psi_x) = n^d C(q), \text{ app-alg} \geq (1 - \delta)n^d C(q). \]
Apply Approx and See What Happens

\( x \in A \) \( \text{MAX3SAT}(\psi_x) = n^d C(q) \), app-alg \( \geq (1 - \delta) n^d C(q) \).

\( x \notin A \) \( \text{MAX3SAT}(\psi_x) \leq n^d (C(q) - 1 + \epsilon) \), so app-alg \( \leq n^d (C(q) - 1 + \epsilon) \).
Apply Approx and See What Happens

\( x \in A \) \( \text{MAX3SAT}(\psi_x) = n^d C(q) \), app-alg \( \geq (1 - \delta)n^d C(q) \).

\( x \notin A \) \( \text{MAX3SAT}(\psi_x) \leq n^d (C(q) - 1 + \epsilon) \), so app-alg \( \leq n^d (C(q) - 1 + \epsilon) \).

For Gap Need

\[
n^d (C(q) - 1 + \epsilon) < (1 - \delta)n^d C(q)
\]

\[
\delta < \frac{1 - \epsilon}{C(q)}
\]
We Won’t Pick $\epsilon$ Cleverly

For Gap Need

\[ \delta < \frac{1 - \epsilon}{C(q)} \]
We Won’t Pick $\epsilon$ Cleverly

For Gap Need

$$\delta < \frac{1 - \epsilon}{C(q)}$$

We want to maximize $\delta$. 

We pick $\epsilon = \frac{1}{4}$, but still call it $\epsilon$.
**We Won’t Pick $\epsilon$ Cleverly**

**For Gap Need**

$$\delta < \frac{1 - \epsilon}{C(q)}$$

We want to maximize $\delta$.

The smaller $\epsilon$ is, the bigger $q$ is, so the bigger $C(q)$ is.
We Won’t Pick $\epsilon$ Cleverly

For Gap Need

$$\delta < \frac{1 - \epsilon}{C(q)}$$

We want to maximize $\delta$.

The smaller $\epsilon$ is, the bigger $q$ is, so the bigger $C(q)$ is.

If we knew how all of these related we would pick $\epsilon$ carefully to maximize $\frac{1 - \epsilon}{C(q)}$. 
We Won’t Pick $\epsilon$ Cleverly

For Gap Need

$$\delta < \frac{1 - \epsilon}{C(q)}$$

We want to maximize $\delta$.

The smaller $\epsilon$ is, the bigger $q$ is, so the bigger $C(q)$ is.

If we knew how all of these related we would pick $\epsilon$ carefully to maximize $\frac{1 - \epsilon}{C(q)}$.

We don’t.
We Won’t Pick \( \epsilon \) Cleverly

For Gap Need

\[
\delta < \frac{1 - \epsilon}{C(q)}
\]

We want to maximize \( \delta \).

The smaller \( \epsilon \) is, the bigger \( q \) is, so the bigger \( C(q) \) is.

If we knew how all of these related we would pick \( \epsilon \) carefully to maximize \( \frac{1-\epsilon}{C(q)} \).

We don’t.

But all we want is there is some \( \delta \) so we can show MAX3SAT has no PTAS.
We Won’t Pick $\epsilon$ Cleverly

For Gap Need

$$\delta < \frac{1 - \epsilon}{C(q)}$$

We want to maximize $\delta$.

The smaller $\epsilon$ is, the bigger $q$ is, so the bigger $C(q)$ is.

If we knew how all of these related we would pick $\epsilon$ carefully to maximize $\frac{1 - \epsilon}{C(q)}$.

We don’t.

But all we want is there is some $\delta$ so we can show MAX3SAT has no PTAS.

We pick $\epsilon = \frac{1}{4}$, but still call it $\epsilon$. 
We Won’t Pick $\epsilon$ Cleverly

For Gap Need

\[ \delta < \frac{1 - \epsilon}{C(q)} \]

We want to maximize $\delta$.

The smaller $\epsilon$ is, the bigger $q$ is, so the bigger $C(q)$ is.

If we knew how all of these related we would pick $\epsilon$ carefully to maximize $\frac{1 - \epsilon}{C(q)}$.

We don’t.

But all we want is there is some $\delta$ so we can show MAX3SAT has no PTAS.

We pick $\epsilon = \frac{1}{4}$, but still call it $\epsilon$.

We pick $\delta = \frac{1 - \epsilon}{2C(q)}$. 
MAX3SAT is Not PTAS: A in P Algorithm

Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \log n, \epsilon)$. Let $C(q)$ be as discussed above.
Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \log n, \epsilon)$.
Let $C(q)$ be as discussed above.
Let $\delta = \frac{1-\epsilon}{2C(q)}$. 
MAX3SAT is Not PTAS: A in P Algorithm

Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \log n, \epsilon)$. Let $C(q)$ be as discussed above.

Let $\delta = \frac{1-\epsilon}{2C(q)}$.

We show there is no $(1 - \delta)$-approx for MAX3SAT. Assume, BWOC, that there is such a app-alg. We use the app-alg, and the PCP, to get $A \in \text{P}$.
Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \lg n, \epsilon)$. Let $C(q)$ be as discussed above. Let $\delta = \frac{1-\epsilon}{2C(q)}$.

We show there is no $(1 - \delta)$-approx for MAX3SAT. Assume, BWOC, that there is such an app-alg. We use the app-alg, and the PCP, to get $A \in P$.

1. Input $x$. 
MAX3SAT is Not PTAS: A in P Algorithm

Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \lg n, \epsilon)$. Let $C(q)$ be as discussed above. Let $\delta = \frac{1 - \epsilon}{2C(q)}$.

We show there is no $(1 - \delta)$-approx for MAX3SAT. Assume, BWOC, that there is such a app-alg. We use the app-alg, and the PCP, to get $A \in \text{P}$.

1. Input $x$.

2. Form the 3CNF formula $\psi_x$. Let $X$ be the number of clauses.
Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \log n, \epsilon)$. Let $C(q)$ be as discussed above.

Let $\delta = \frac{1 - \epsilon}{2C(q)}$.

We show there is no $(1 - \delta)$-approx for MAX3SAT. Assume, BWOC, that there is such an app-alg. We use the app-alg, and the PCP, to get $A \in P$.

1. Input $x$.
2. Form the 3CNF formula $\psi_x$. Let $X$ be the number of clauses.
3. Apply the approx to $\psi_x$. Call the result $Y$. 

By the commentary in the last few slides, and the choice of $\delta$, exactly one of the inequalities for $Y$ holds.
MAX3SAT is Not PTAS: A in P Algorithm

Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \lg n, \epsilon)$. Let $C(q)$ be as discussed above.

Let $\delta = \frac{1 - \epsilon}{2C(q)}$.

We show there is no $(1 - \delta)$-approx for MAX3SAT. Assume, BWOC, that there is such a app-alg. We use the app-alg, and the PCP, to get $A \in \mathbb{P}$.

1. Input $x$.
2. Form the 3CNF formula $\psi_x$. Let $X$ be the number of clauses.
3. Apply the approx to $\psi_x$. Call the result $Y$.
4. If $Y \geq (1 - \delta)n^d C(q)$ then output YES, $x \in A$. 

By the commentary in the last few slides, and the choice of $\delta$, exactly one of the inequalities for $Y$ holds.
MAX3SAT is Not PTAS: A in P Algorithm

Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \lg n, \epsilon)$. Let $C(q)$ be as discussed above.
Let $\delta = \frac{1 - \epsilon}{2C(q)}$.

We show there is no $(1 - \delta)$-approx for MAX3SAT.
Assume, BWOC, that there is such a app-alg. We use the app-alg, and the PCP, to get $A \in \mathbb{P}$.

1. Input $x$.

2. Form the 3CNF formula $\psi_x$. Let $X$ be the number of clauses.

3. Apply the approx to $\psi_x$. Call the result $Y$.

4. If $Y \geq (1 - \delta)n^dC(q)$ then output YES, $x \in A$.
5. If $Y \leq n^d(C(q) - 1 + \epsilon)$ then output NO, $x \notin A$. 
MAX3SAT is Not PTAS: A in P Algorithm

Let $\epsilon = \frac{1}{4}$. Let $q, d$ be such that $A \in \text{PCP}(q, d \lg n, \epsilon)$. Let $C(q)$ be as discussed above.

Let $\delta = \frac{1 - \epsilon}{2C(q)}$.

We show there is no $(1 - \delta)$-approx for MAX3SAT.
Assume, BWOC, that there is such a app-alg. We use the app-alg, and the PCP, to get $A \in \mathbb{P}$.

1. Input $x$.
2. Form the 3CNF formula $\psi_x$. Let $X$ be the number of clauses.
3. Apply the approx to $\psi_x$. Call the result $Y$.
4. If $Y \geq (1 - \delta)n^dC(q)$ then output YES, $x \in A$.
5. If $Y \leq n^d(C(q) - 1 + \epsilon)$ then output NO, $x \notin A$.

By the commentary in the last few slides, and the choice of $\delta$, exactly one of the inequalities for $Y$ holds.