IP = PSPACE.

First step: prove $coNP \subseteq IP$.

Recall randomness is needed for full power of IP and one of few problems in BPP but not P is PIT. Let us try converting $\overline{3SAT}$ into a problem about polynomials.

We think of T/F as 1/0. Then
$\vee \to +$
$\wedge \to \cdot$
$\neg \to (1-)$.

e.g. $(x_3 \vee \overline{x_5} \vee x_{17}) \wedge (x_5 \vee x_9) \wedge (\overline{x_3} \vee \overline{x_4})$

$$\downarrow$$

$$(x_3 + (1-x_5) + x_{17}) \cdot (x_5 + x_9) \cdot ((1-x_3) + (1-x_4)).$$

obs $F(\alpha)$ true $\to p(\alpha) > 0$ $\forall \alpha$.
$F(\alpha)$ false $\to p(\alpha) = 0$

hence
$$\sum_{\alpha \in \{0,1\}^n} p(\alpha) = \begin{cases} 0 & F \text{ unsat} \\ >0 & F \text{ sat.} \end{cases}$$

obs $\deg(p) = m = |F|$ // i.e. #clauses.

obs $p(\alpha) \leq 3^m$. (hence $\sum_\alpha p(\alpha) \leq 2^n \cdot 3^m$)
$\downarrow$
poly many bits.

so we'd like to distinguish $p(\alpha) = 0$ or $\in [1, 2^n \cdot 3^m]$. convenient to work on a field $\mathbb{F}_\ell$, ℓ prime $> 2^n \cdot 3^m$.

(prover can choose ℓ and verifier check deterministically). then can do PIT and eval $\sum_\alpha p(\alpha)$ on a random point $\alpha \in \mathbb{F}_\ell^n$. Obs $\mathbb{F}_\ell$ and not $\{0,1\}$ helps a lot: if $\alpha \in \{0,1\}^n$ there could be a unique sat. α and we'd only catch it with pr. 2^{-n}. But in $\mathbb{F}_\ell$ we can use Schwartz-Zippel-deMillo-Lipton-....
Except not really because $\sum p$ is too large. instead, we'll replace variables one by one.

Plan: iteratively certify $\sum_{x_i=0}^{1} \cdots \sum_{x_n=0}^{1} p(r_1 \ldots r_{i-1}, x_i \ldots x_n) = v_{i-1}$.

def $a_i(x_1 \ldots x_i) = \sum_{x_{i+1}} \cdots \sum p(x_1 \ldots x_n)$.

obs. $a_i(—) = a_{i+1}(—0) + a_{i+1}(—1)$.

So our plan is: prove that
$\exists \{a_i\}_i$ st $a_i = a_{i+1}(0—) + a_{i+1}(1—)$.

and $a_0 = 0$
and $a_n = p$. but a_i too expensive to send.

instead we'll prove

$a_i(r_1 \ldots r_i) = a_{i+1}(r_1 \ldots r_i\, 0) + a_{i+1}(r_1 \ldots r_i\, 1)$.
i.o.w. $\exists\, q_{i+1}(z) = a_{i+1}(r_1 \ldots r_i, x_{i+1} = z)$
st $q_{i+1}(0) + q_{i+1}(1) = a_i(r_1 \ldots r_i) = v_i$.
obs $q(z)$ single-variable and degree m, hence P can compute q and send to V.
then V evals q on $0,1$; randomly picks r_i and sends to prover; and both set $v_i = q(r_i)$.
First test is $q(0) + q(1) = 0$.
Last test is $q(r_n) = p(r_1 \ldots r_n)$.

Formally:
$v_0 = 0$.
for $i = 1..n$: P sends q'
V checks $q'(0) + q'(1) = v_{i-1}$ and $\deg q' \leq m$.
V samples & sends r_i
$v_i = q'(r_i)$.
V checks $v_n = p(r_1 \ldots r_n)$.

If $\sum_\alpha p(\alpha) = 0$ then exists q' st P always accepts (pick $q' = q$).

claim: If $\sum_\alpha p(\alpha) \neq 0$ then P rejects whp.

Assume wlog in i-th round P sends q' of deg $\leq m$, and st $q'(0) + q'(1) = v_{i-1}$ (o/w V notices).
If fail then there is some round where
$v_{i-1} = a(r_1 \ldots r_{i-1})$ but $v_i = a(r_1 \ldots r_i)$.
we have $q'(0) + q'(1) = v_{i-1} \Rightarrow q' \neq q$.
Then $q' - q$ is a poly of deg $\leq m$ and has $\leq m$ roots; $\Pr[r_i \text{ root}] \leq (1 - \frac{m}{\ell})$.

This proves $coNP \subseteq IP$. Now let us see $PSPACE \subseteq IP$.

Like before, translate QBF $\exists x_1 \forall x_2 \ldots F(\cdot)$. into

$\sum_{x_1=0}^{1} \prod_{x_2=0}^{1} \cdots p(\cdot)$; want to know if 0 or >0.

Problem: each Π may square the polynomial, so q could have degree $m^{n/2}$, too expensive for prover to send.

Recall even if we're evaluating $p(\cdot)$ on weird points, we only care about 0/1.
So we are allowed to evaluate any polynomial
$p' \equiv p \mod (x_1 = 0 \vee x_1 = 1,\ x_2 = 0 \vee x_2 = 1,\ \ldots\ x_n = 0 \vee x_n = 1)$.
iow mod ideal $I = \langle x_1(x_1 - 1), x_2(x_2 - 1), \ldots x_n(x_n - 1) \rangle$.
iow we can replace $x_i^2 \mapsto x_i$, and more generally $x_i^d \mapsto x_i$.
Let us define L_i as the multilinearization operator.
$L_i(p) = (1 - x_i) \cdot p(x_1 \ldots x_{i-1}\, 0\, x_{i+1} \ldots x_n) + x_i \cdot p(—1—)$.
Let us also define $S_i(p) = p(—0—) + p(—1—)$.
$P_i(p) = p(—0—) \cdot p(—1—)$.

We want to know if
$S_1 P_2 \ldots p = 0$ or >0.
equiv.
$S_1 L_1 P_2 L_1 L_2 \ldots p = 0$ // obs n^2 operators.

we can run same protocol as before.

$v_0 = 0$.
for $i = 1..n^2$: P sends q'
V checks $\deg q' \leq m^2$.
if op is S_i: check $q'(0) + q'(1) = v_{i-1}$.
if op is P_i: check $q'(0) \cdot q'(1) = v_{i-1}$.
if op is L_i: check $r_i \cdot q'(0) + (1 - r_i) q'(1) = v_{i-1}$.
V samples & sends r_i (replacing previous r_i if needed).
$v_i = q'(r_i)$
V checks $v_{n^2} = p(r_1 \ldots r_n)$.