SIKE (in Round 2)

Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, David Jao, Brian Koziel, Geovandro Pereira, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev

amazon

UNIVERSITY OF WATERLOO

UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES

TEXAS INSTRUMENTS

Microsoft Research

Radboud University

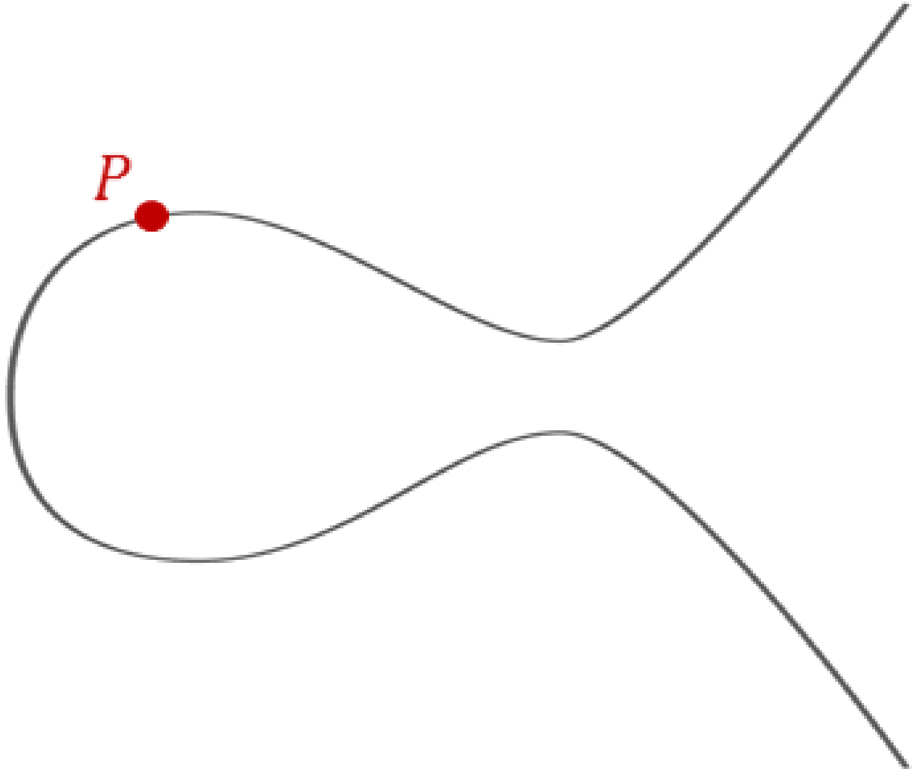
FAU FLORIDA ATLANTIC UNIVERSITY

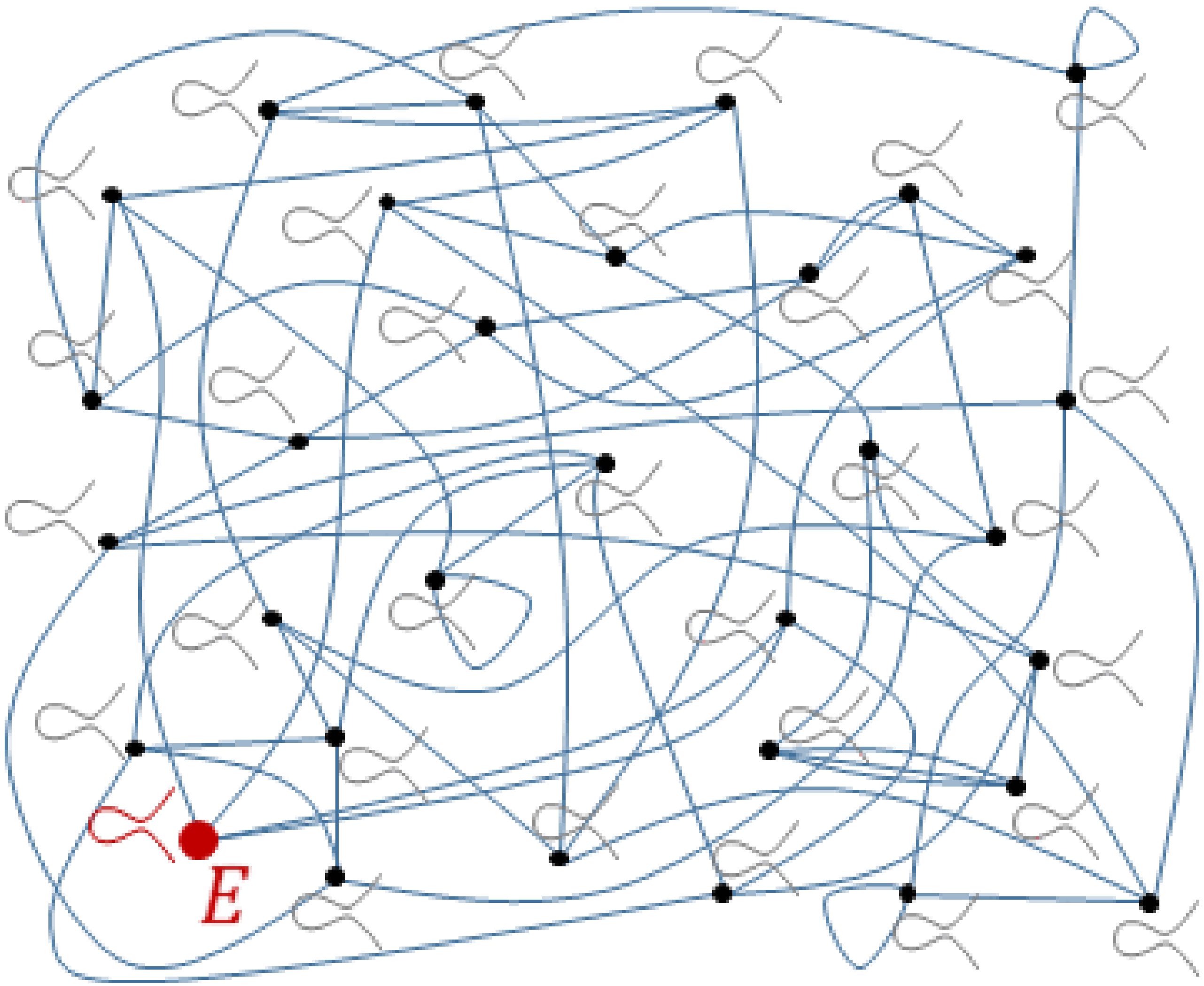
March 20, 2019
Oxford PQC Workshop
Oxford, UK

SIKE Round 2 updates

- Smaller parameters: attacks are worse in practice
- Compression: even smaller public keys / ciphertexts
- New starting curve: *a bit* better

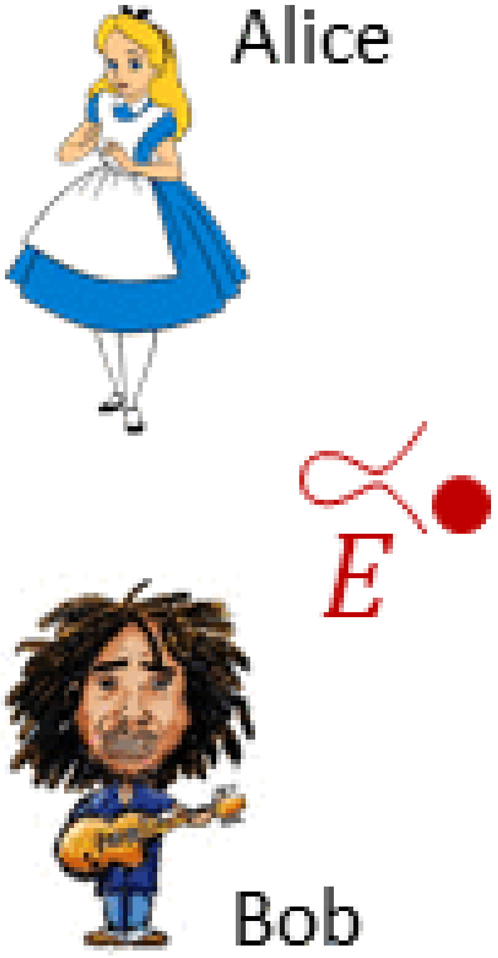
ECC vs. post-quantum ECC





W. Castryck (GIF): https://www.esat.kuleuven.be/cosic/?p=7404

Alice 2^e-isogenies, Bob 3^f-isogenies



W. Castryck (GIF): https://www.esat.kuleuven.be/cosic/?p=7404

Diffie-Hellman instantiations

	DH	ECDH	SIDH/SIKE
Elements	integers g modulo prime	points P in curve group	curves E in isogeny class
Secrets	exponents x	scalars k	isogenies ϕ
computations	$g, x \mapsto g^x$	$k, P \mapsto [k]P$	$\phi, E \mapsto \phi(E)$
hard problem	given g, g^x find x	given $P, [k]P$ find k	given $E, \phi(E)$ find ϕ

SIDH/SIKE setup

$$p = 2^i \cdot 3^j - 1$$

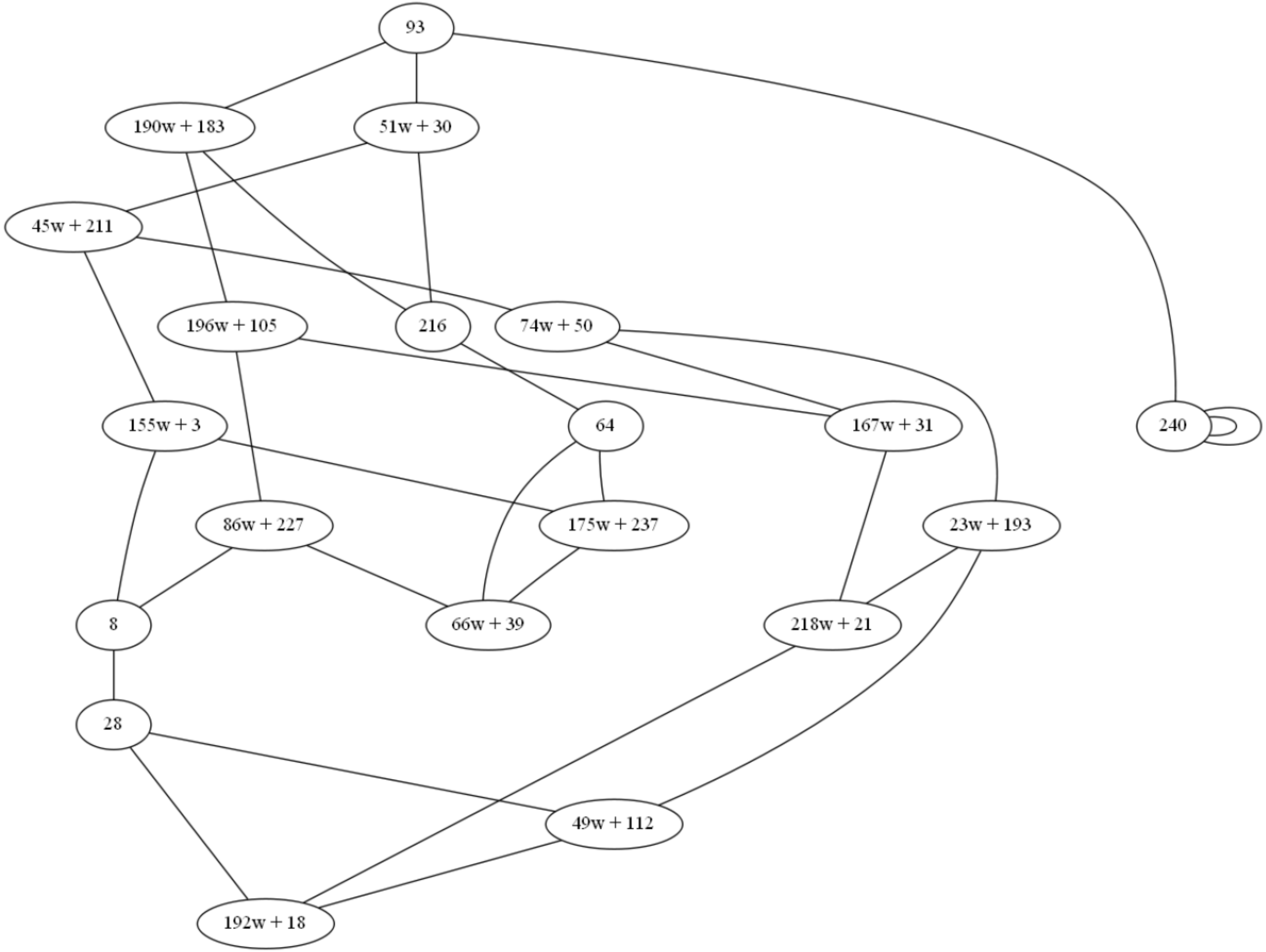
- Elements are supersingular elliptic curves over $\mathbb{F}_{p^2}$ (up to $\cong$)

- Roughly $p/12$ of them

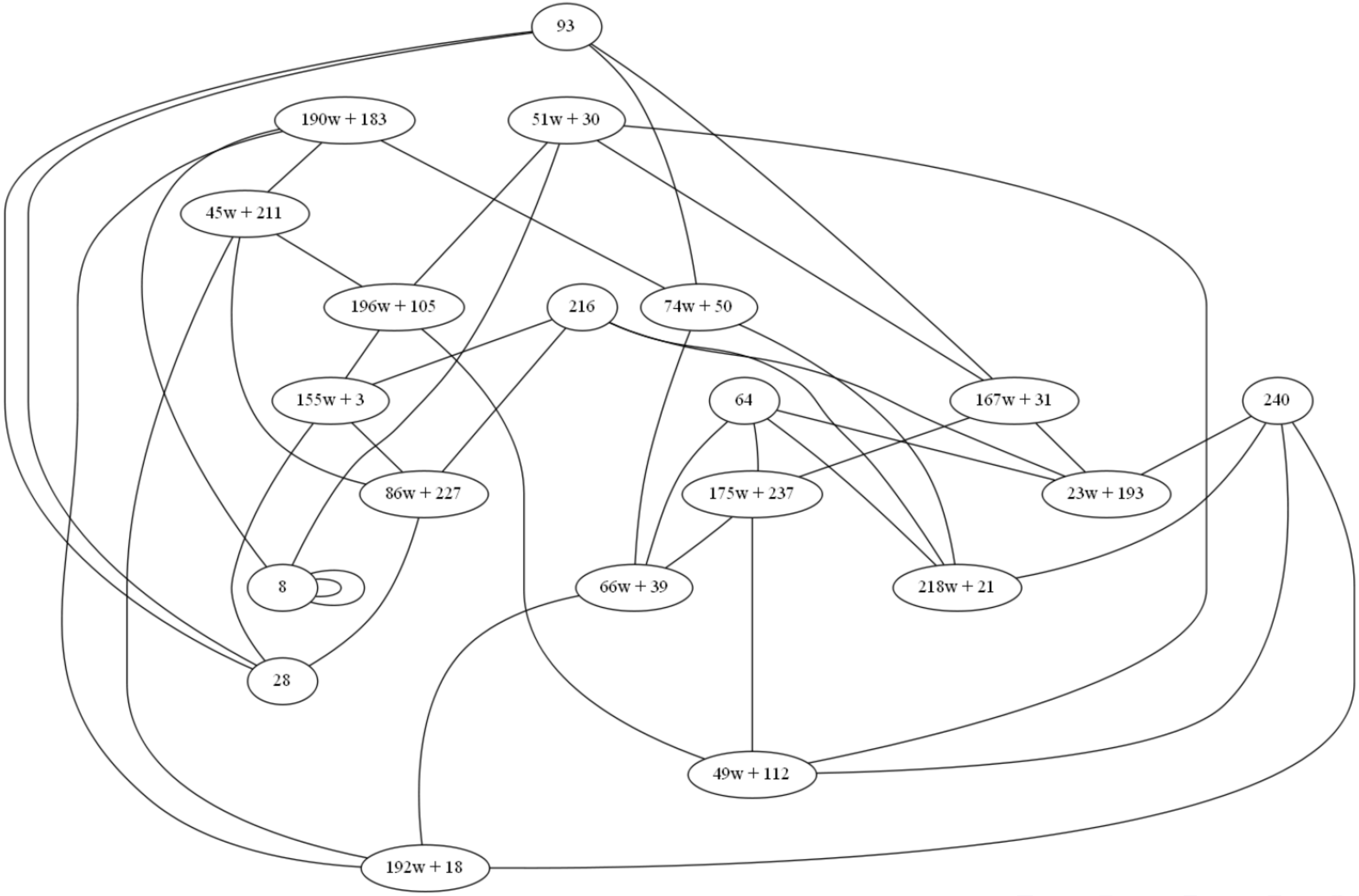
- For any ℓ (not a multiple of p), set forms a $(\ell + 1)$-regular graph that is Ramanujan: edges are isogenies, $\ell \in \{2,3\}$ means they're $\mathbb{F}_{p^2}$-rational

- Easiest with an example…

Supersingular isogeny graph for $\ell = 2$: $X(S_{241^2}, 2)$



Supersingular isogeny graph for $\ell = 3$: $X(S_{241^2}, 3)$



Cyclic subgroup isogenies

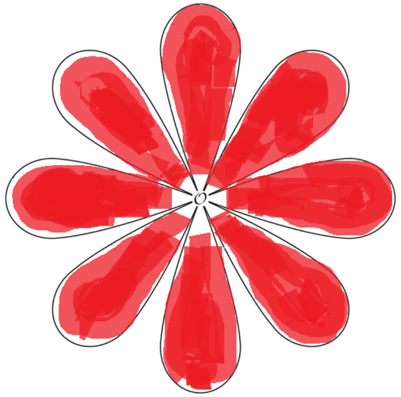
- Maps $\phi : E \to E'$ that are (algebraic/geometric) morphisms

$$(x, y) \mapsto (x', y')$$

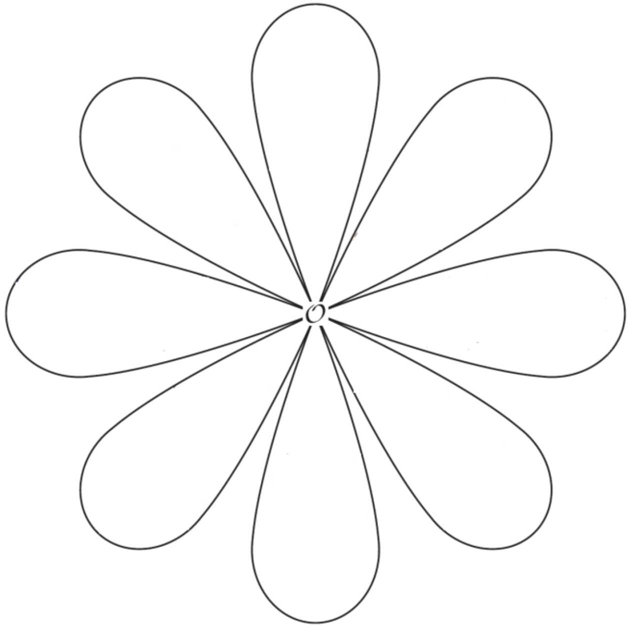
- Similar to (e.g.) multiplication-by-n, except we land on a different curve

$E[n] \cong \mathbb{Z}_n \times \mathbb{Z}_n$

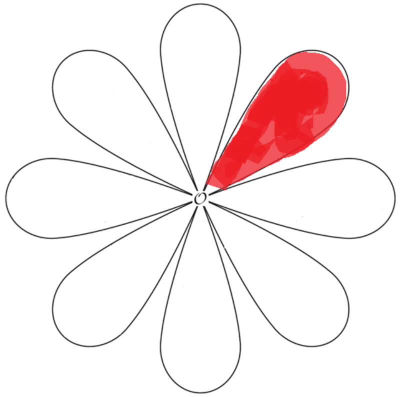
Kernel of $[n]$
$\cong \mathbb{Z}_n \times \mathbb{Z}_n$



Degree is n^2



Kernel of cyclic n-isogeny
$\cong \mathbb{Z}_n$



Degree is n

E.g. Montgomery $\mathbf{2}$-isogeny

$$E: \; y^2 = x^3 + Ax^2 + x \qquad\qquad E': \; y^2 = x^3 + A'x^2 + x$$

$E[2] = \{O_E, (0{,}0), (\alpha, 0), (1/\alpha, 0)\}$

$$[2]: E \to E, \qquad x \mapsto \frac{\left(x^2-1\right)^2}{4x(x^2+Ax+x)} \qquad \ker([2]) = E[2]$$

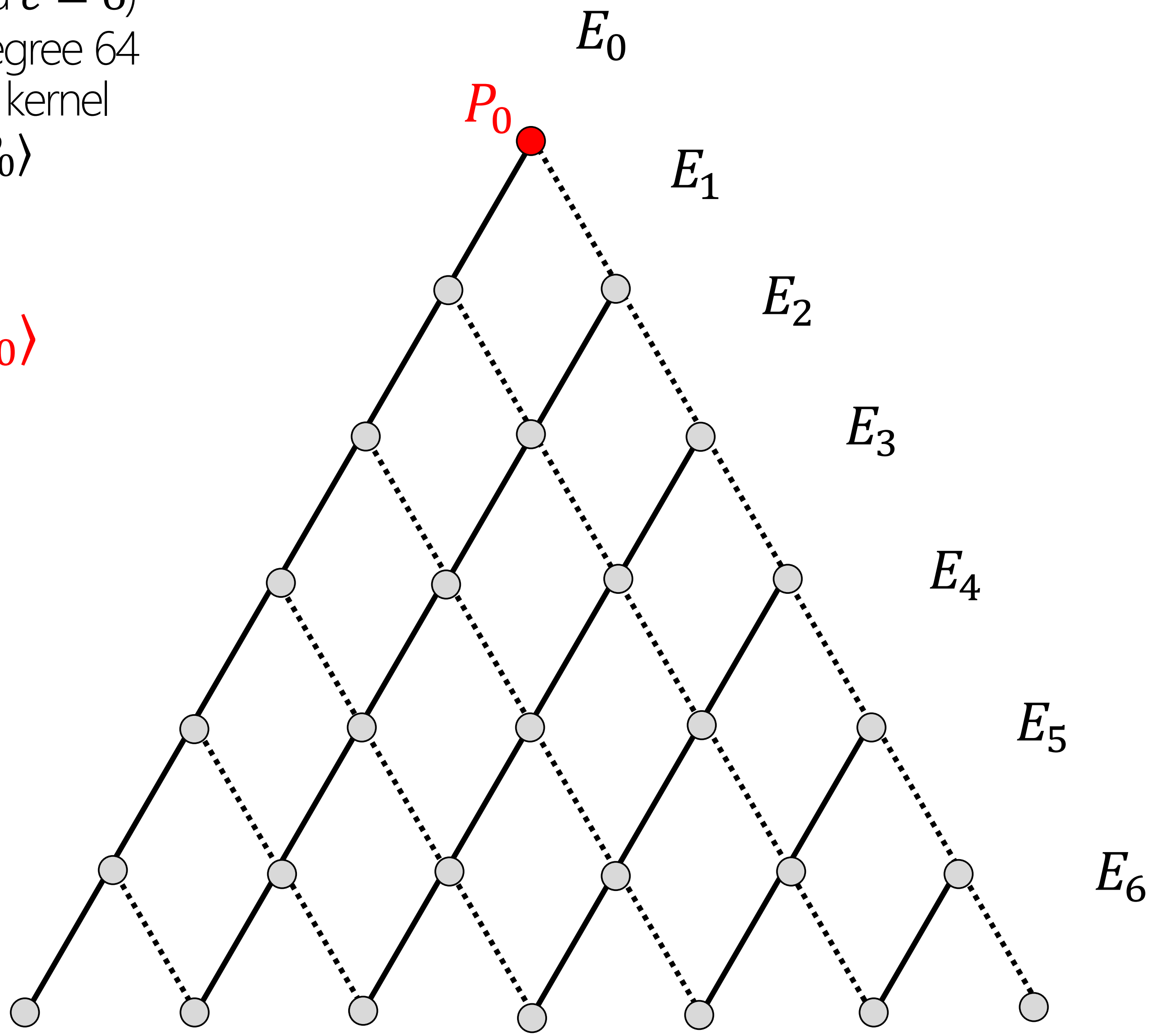
$$\phi: E \to E', \qquad x \mapsto x \cdot \left(\frac{\alpha x - 1}{x - \alpha}\right) \qquad \ker(\phi) = \{O_E, (\alpha, 0)\}$$

In practice we work entirely in $\mathbb{P}^1$, i.e., $(X:Z) \mapsto (X':Z')$, etc

Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : \ E_0 \rightarrow E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_6 = E_0/\langle P_0 \rangle$



Computing ℓ^e degree isogenies

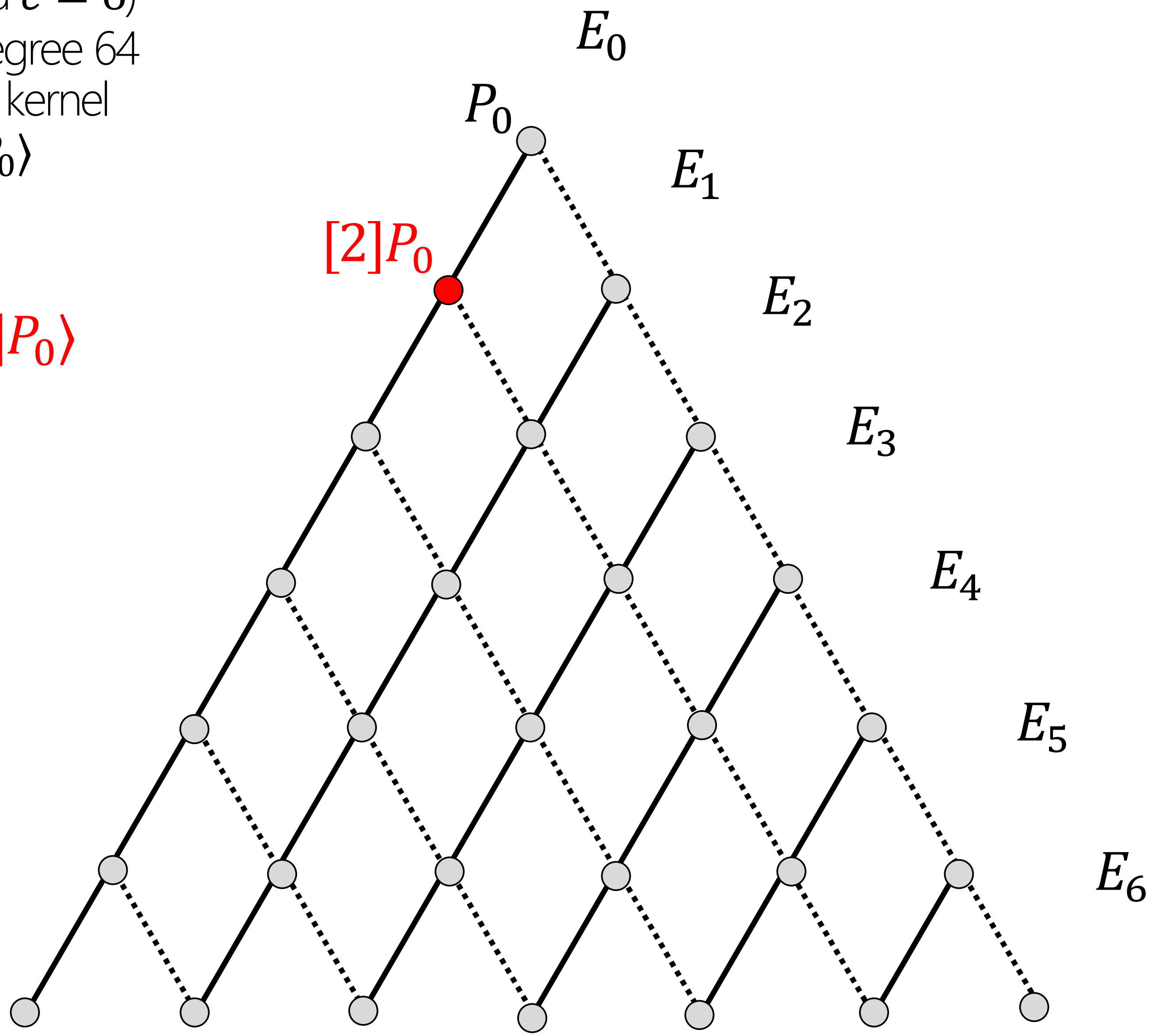
(suppose $\ell = 2$ and $e = 6$)

$\phi : \ E_0 \to E_6$ is degree 64

64 elements in its kernel

$\ker(\phi) = \langle P_0 \rangle$

$E_5 = E_0 / \langle [2]P_0 \rangle$



Computing ℓ^e degree isogenies

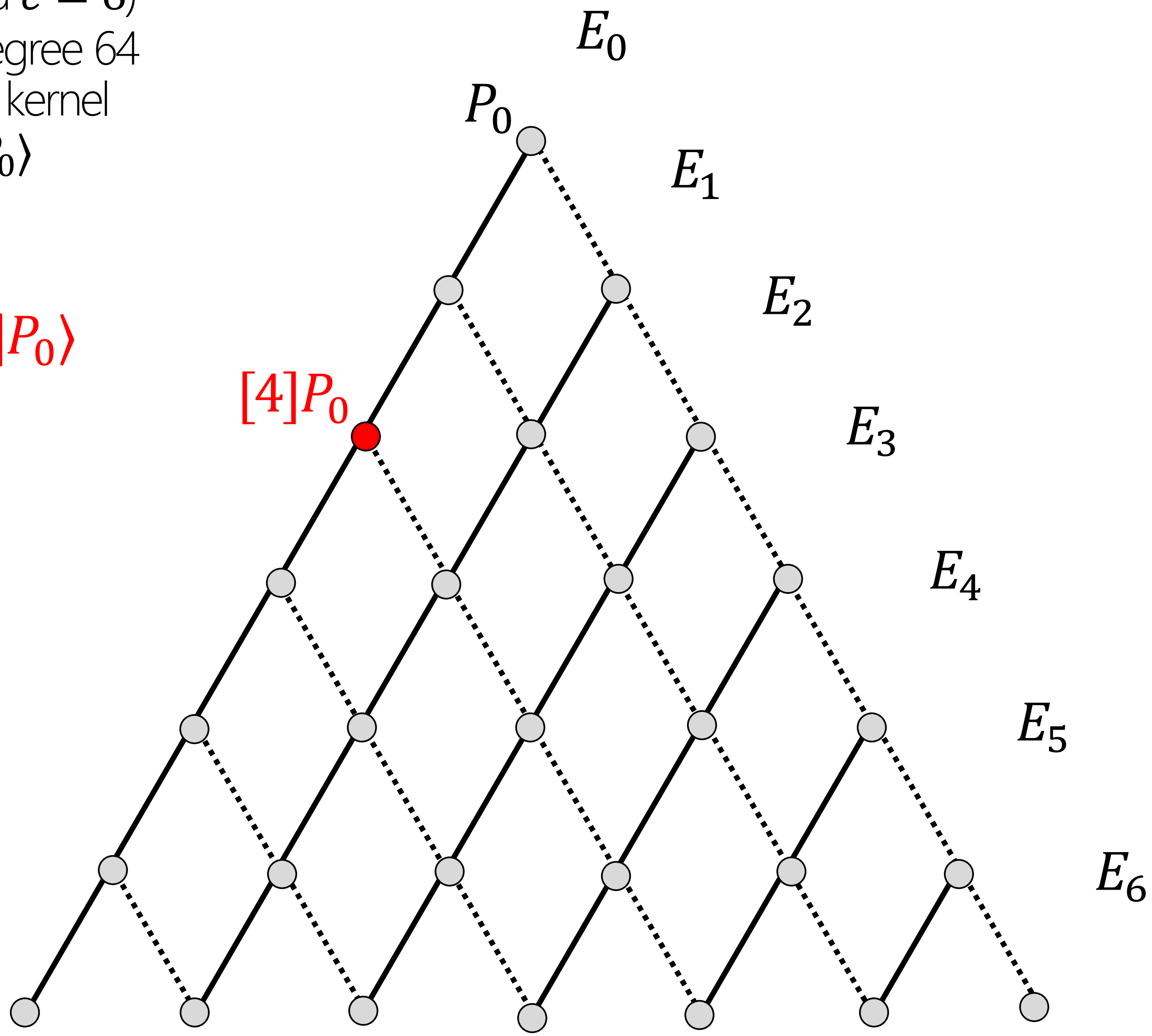
(suppose $\ell = 2$ and $e = 6$)

$\phi : \ E_0 \rightarrow E_6$ is degree 64

64 elements in its kernel

$\ker(\phi) = \langle P_0 \rangle$

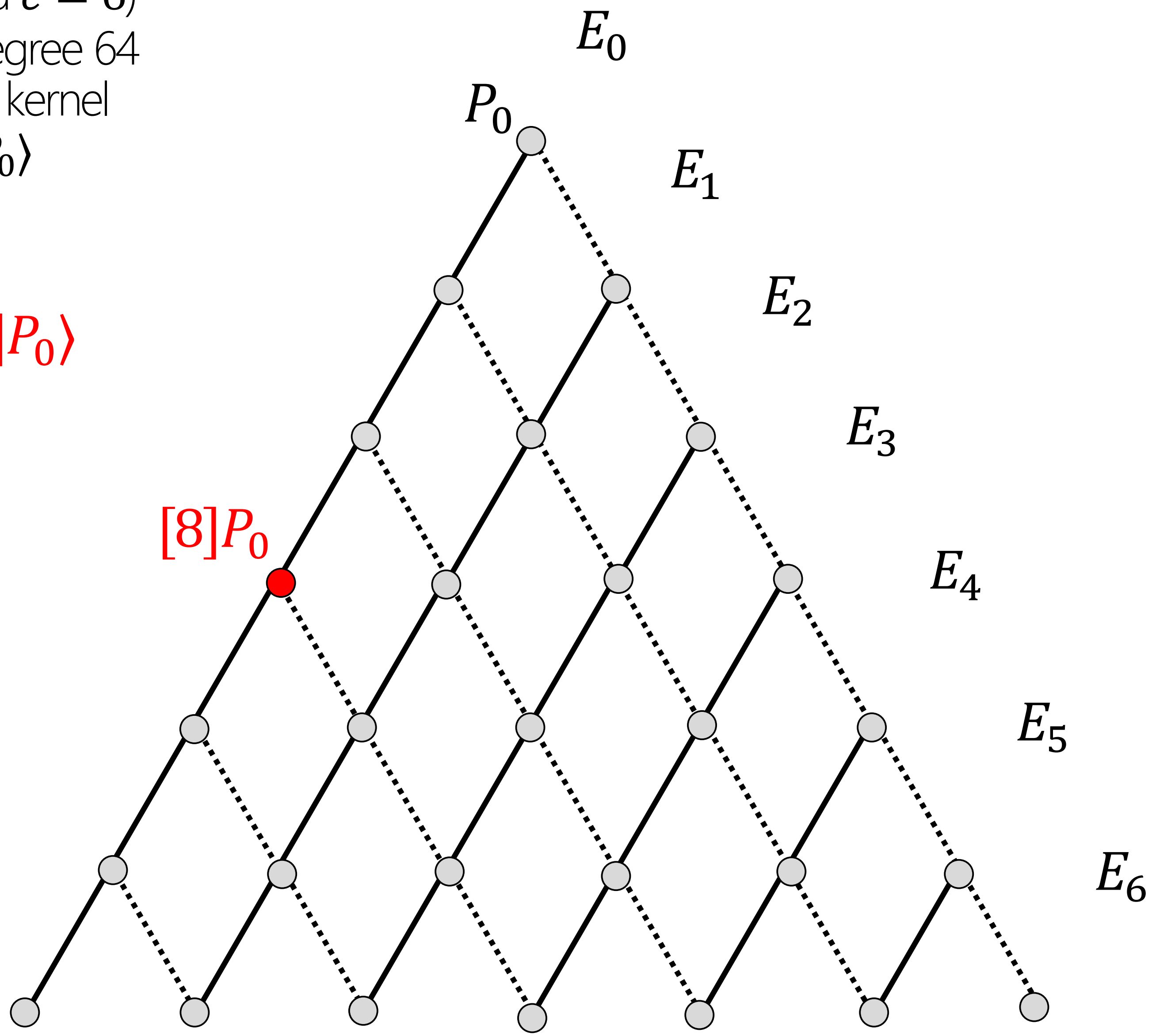
$E_4 = E_0 / \langle [4]P_0 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

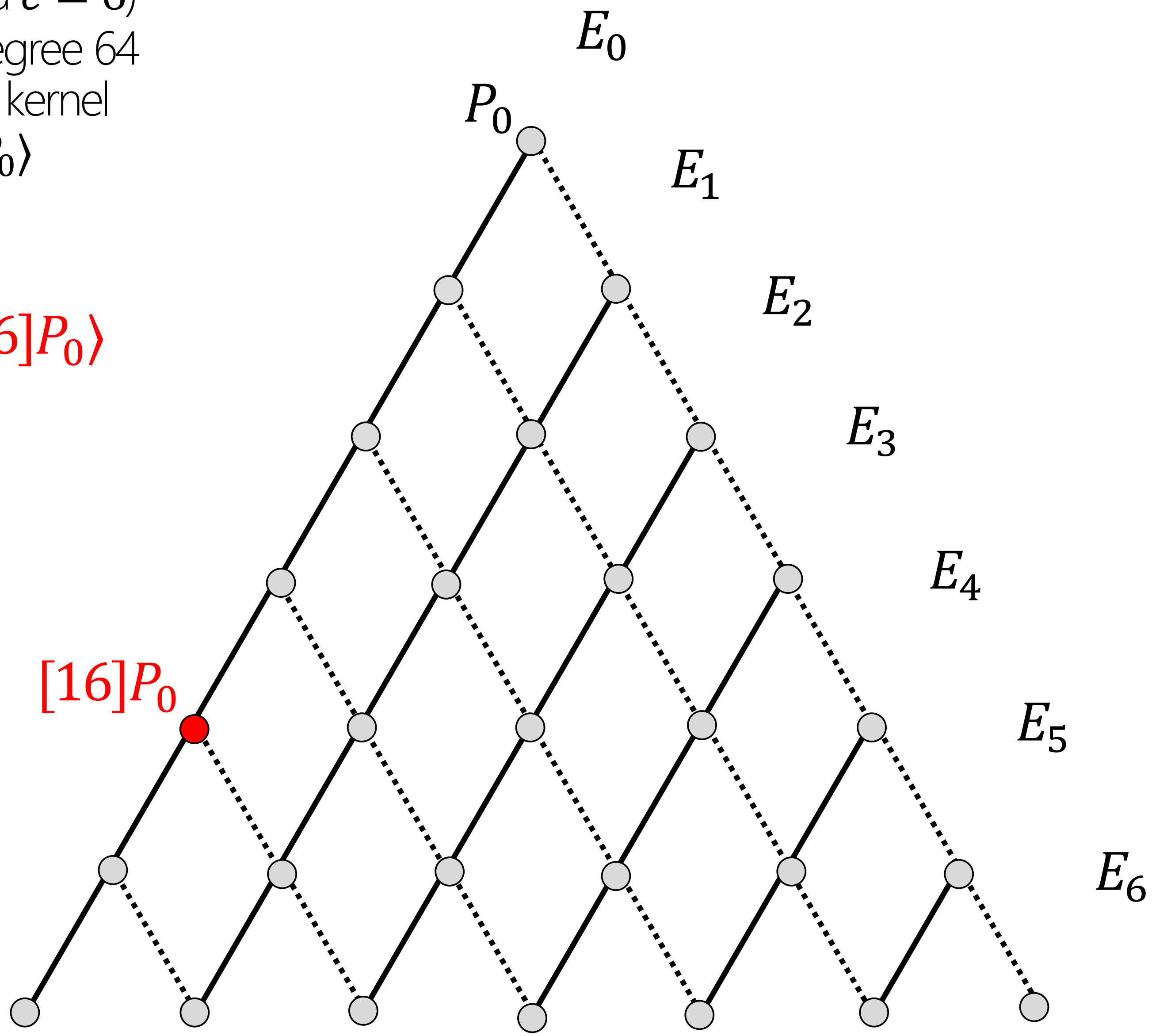
$E_3 = E_0/\langle [8]P_0 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : \ E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

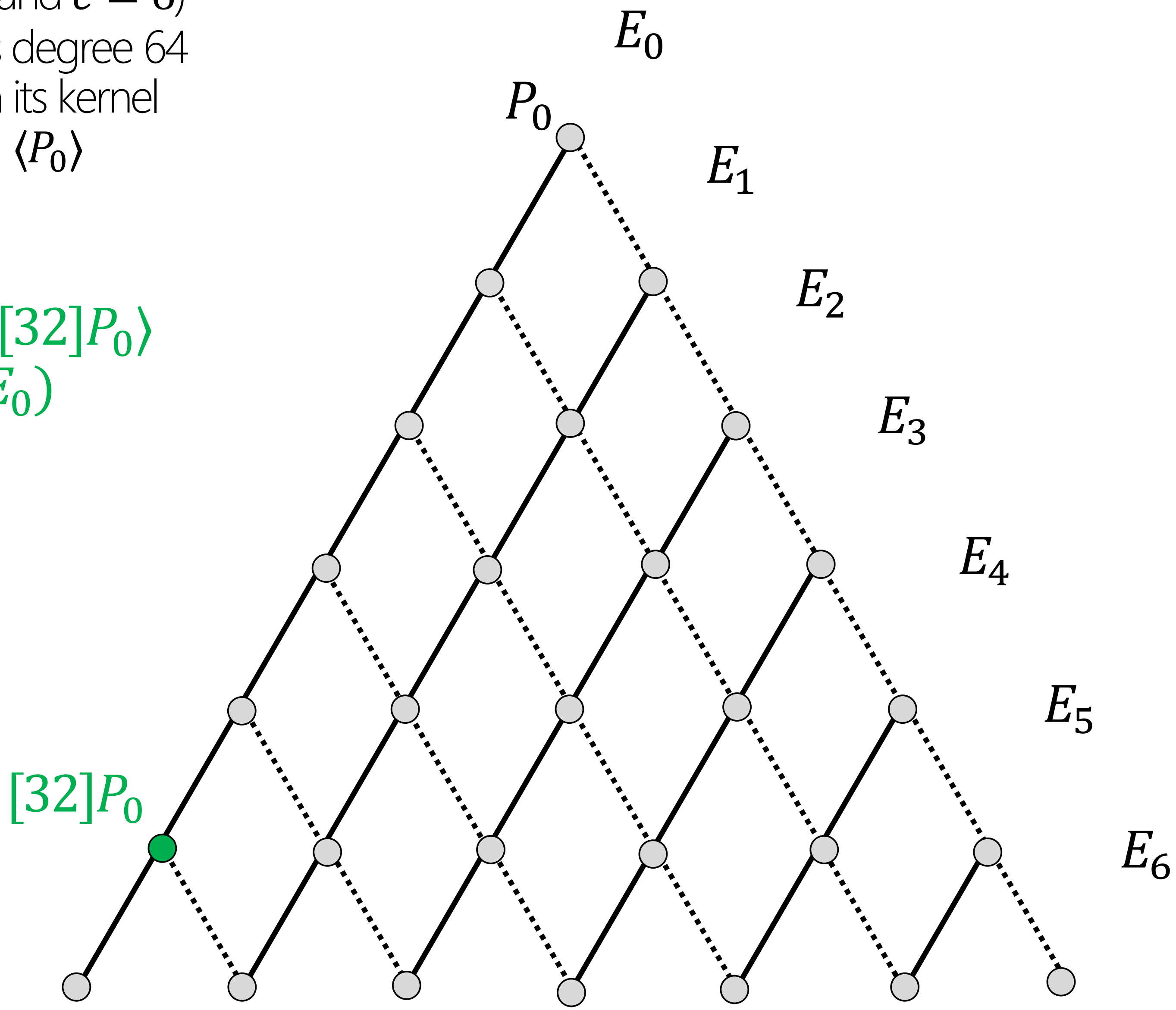
$E_2 = E_0 / \langle [16]P_0 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_1 = E_0/\langle [32]P_0 \rangle$
$= \phi_0(E_0)$

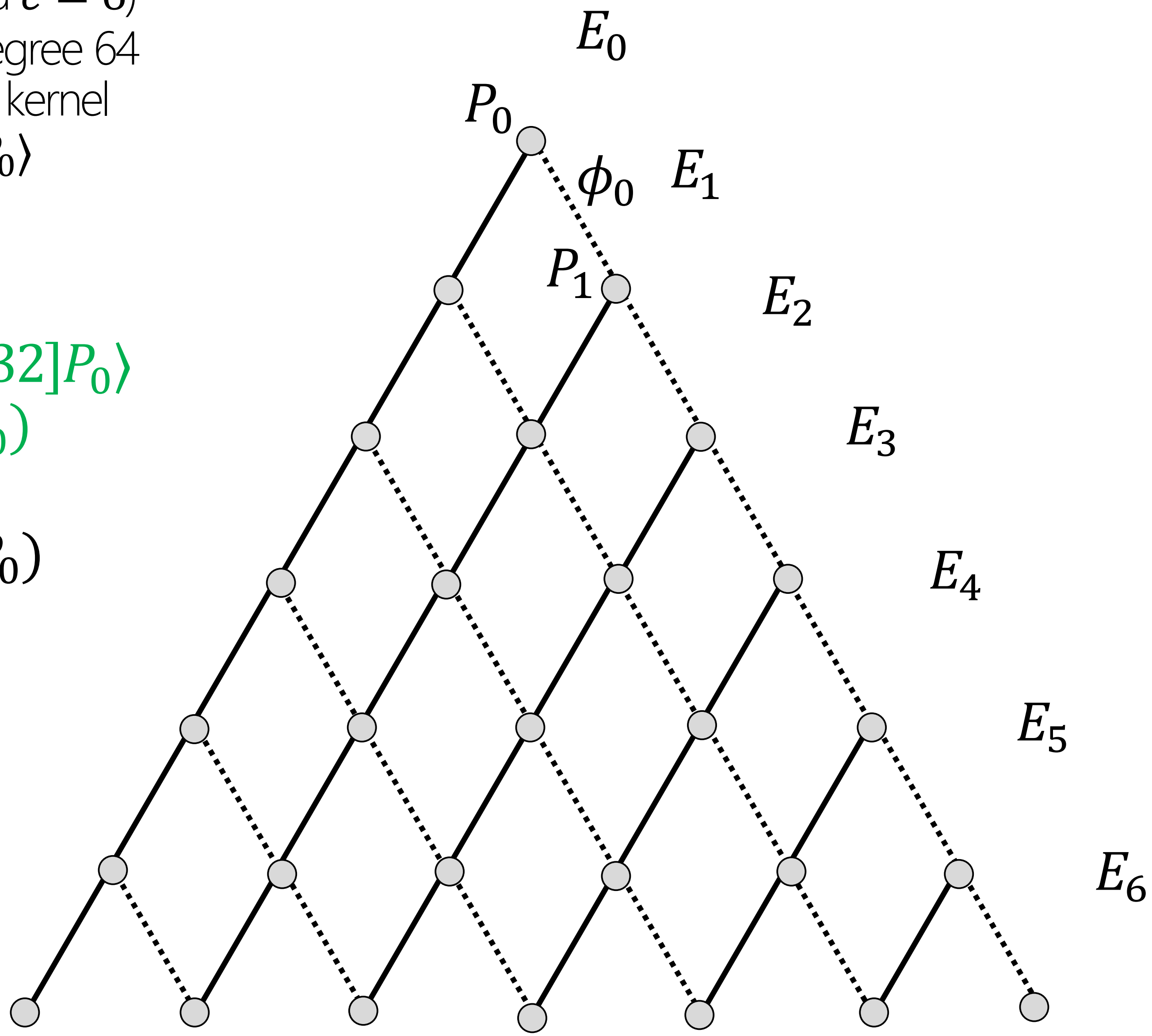


Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_1 = E_0/\langle [32]P_0 \rangle$
$= \phi_0(E_0)$

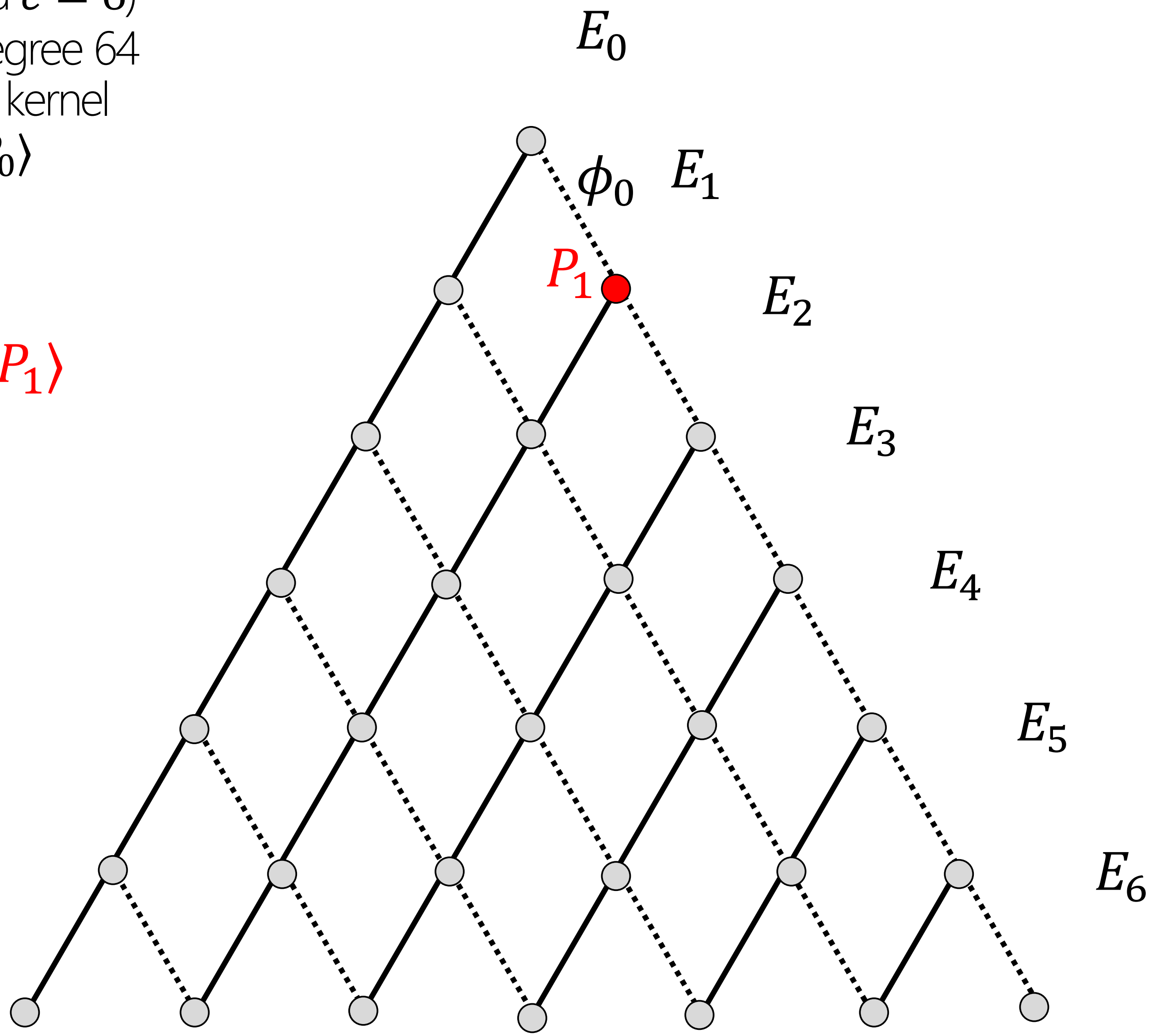
$P_1 = \phi_0(P_0)$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi: \; E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

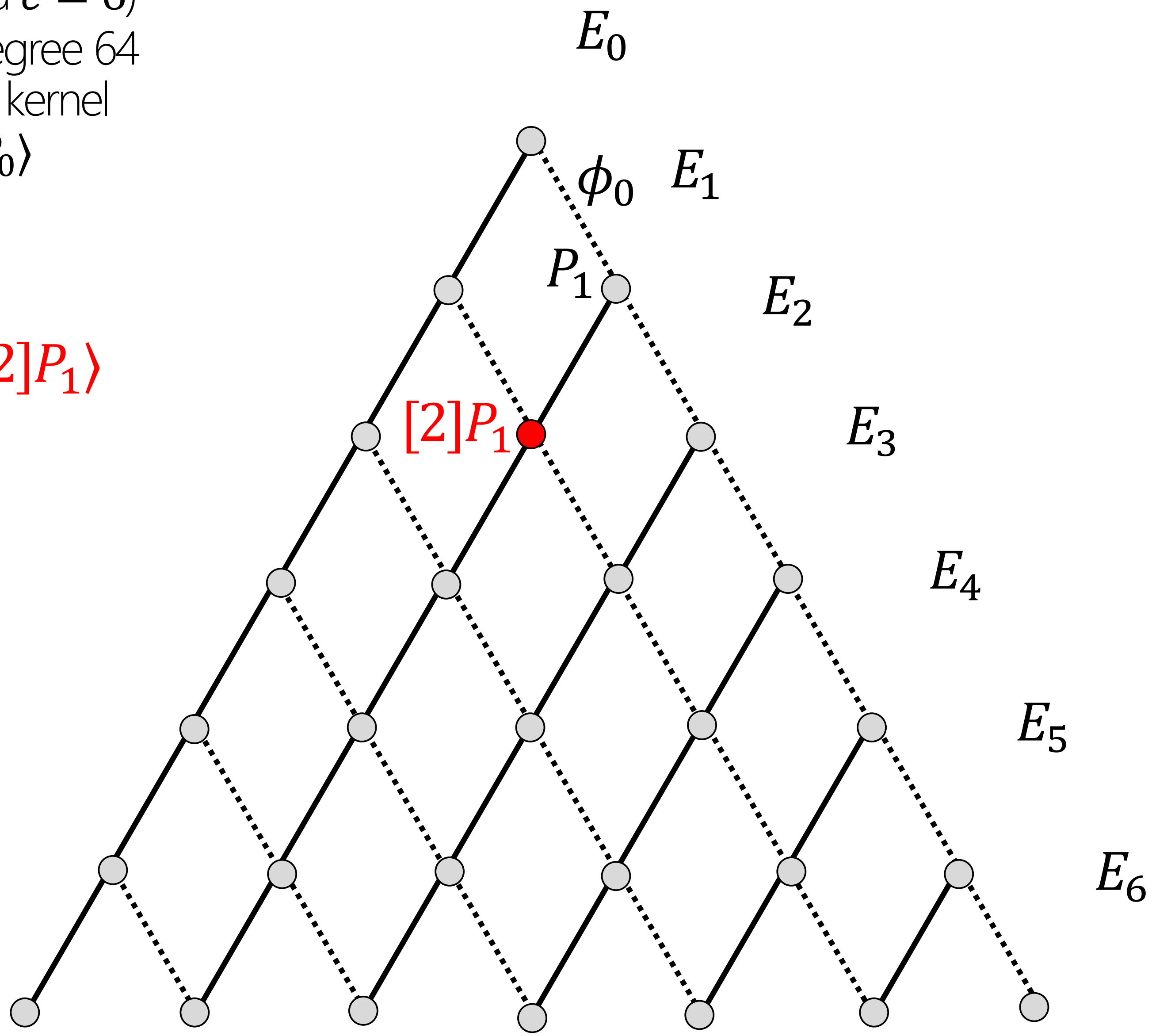
$E_6 = E_1/\langle P_1 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi: E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

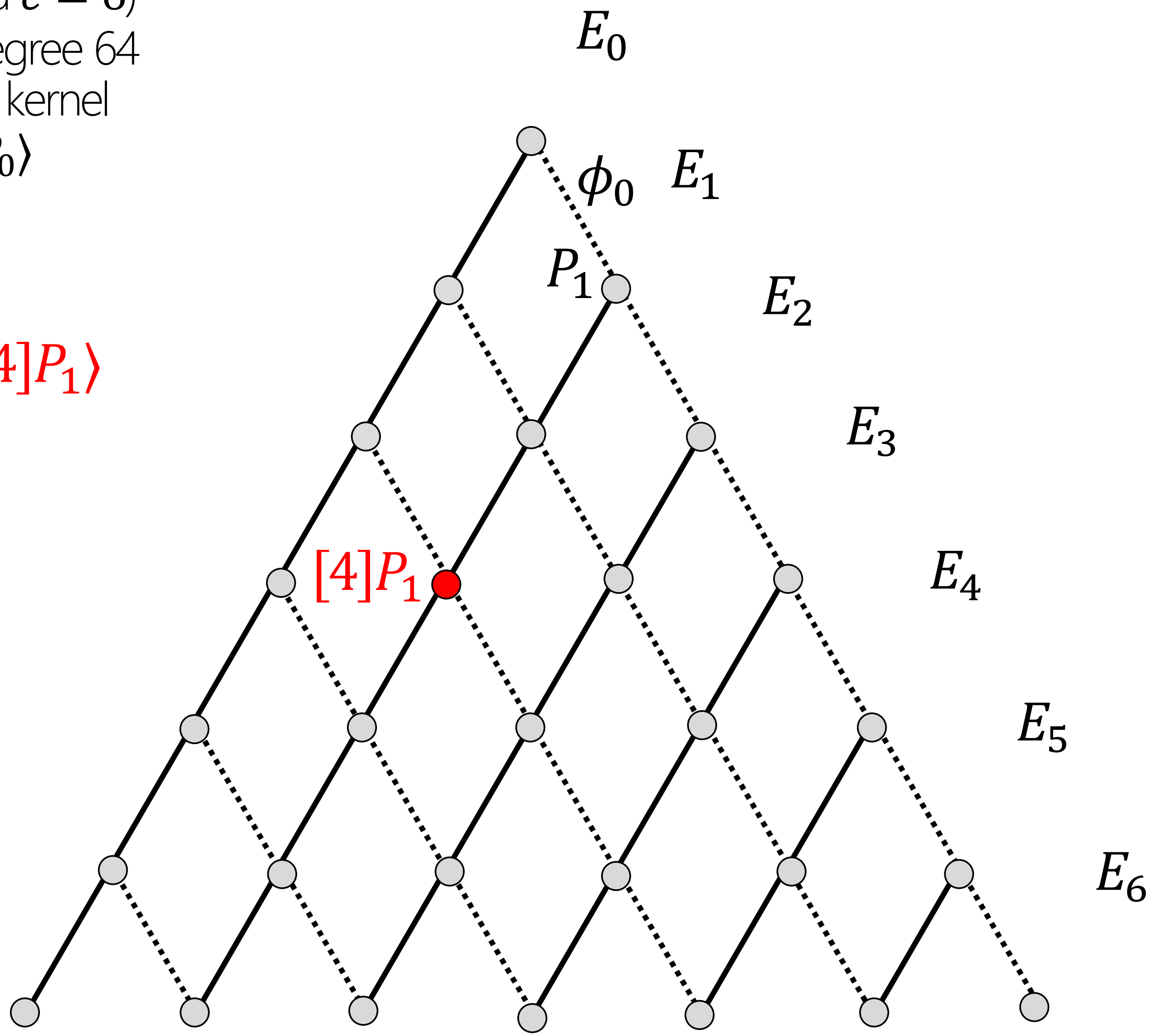
$E_5 = E_1/\langle [2]P_1 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi :\ E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

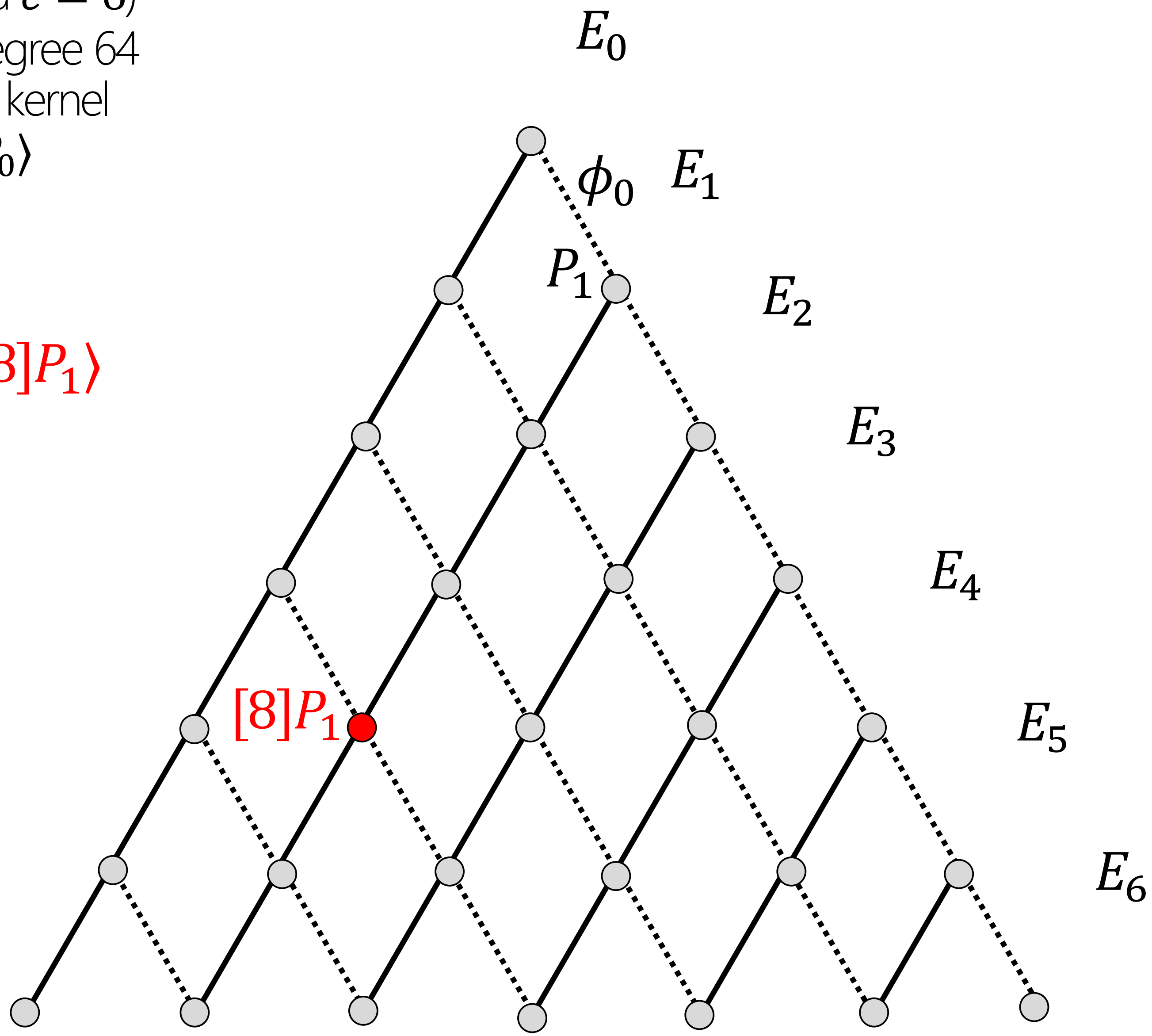
$E_4 = E_1/\langle [4]P_1 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

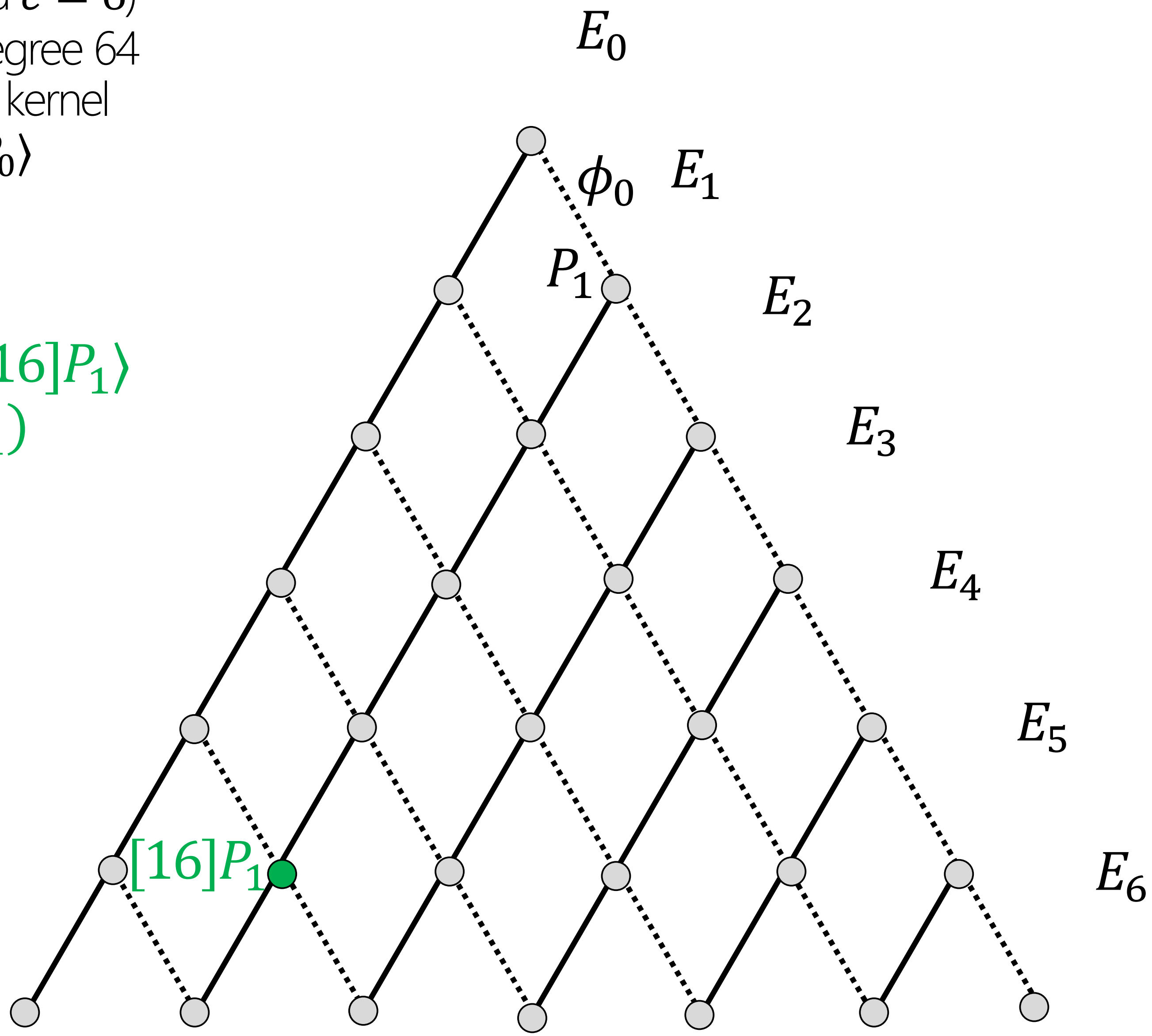
$E_3 = E_1 / \langle [8]P_1 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : \; E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_2 = E_1/\langle [16]P_1 \rangle$
$= \phi_1(E_1)$

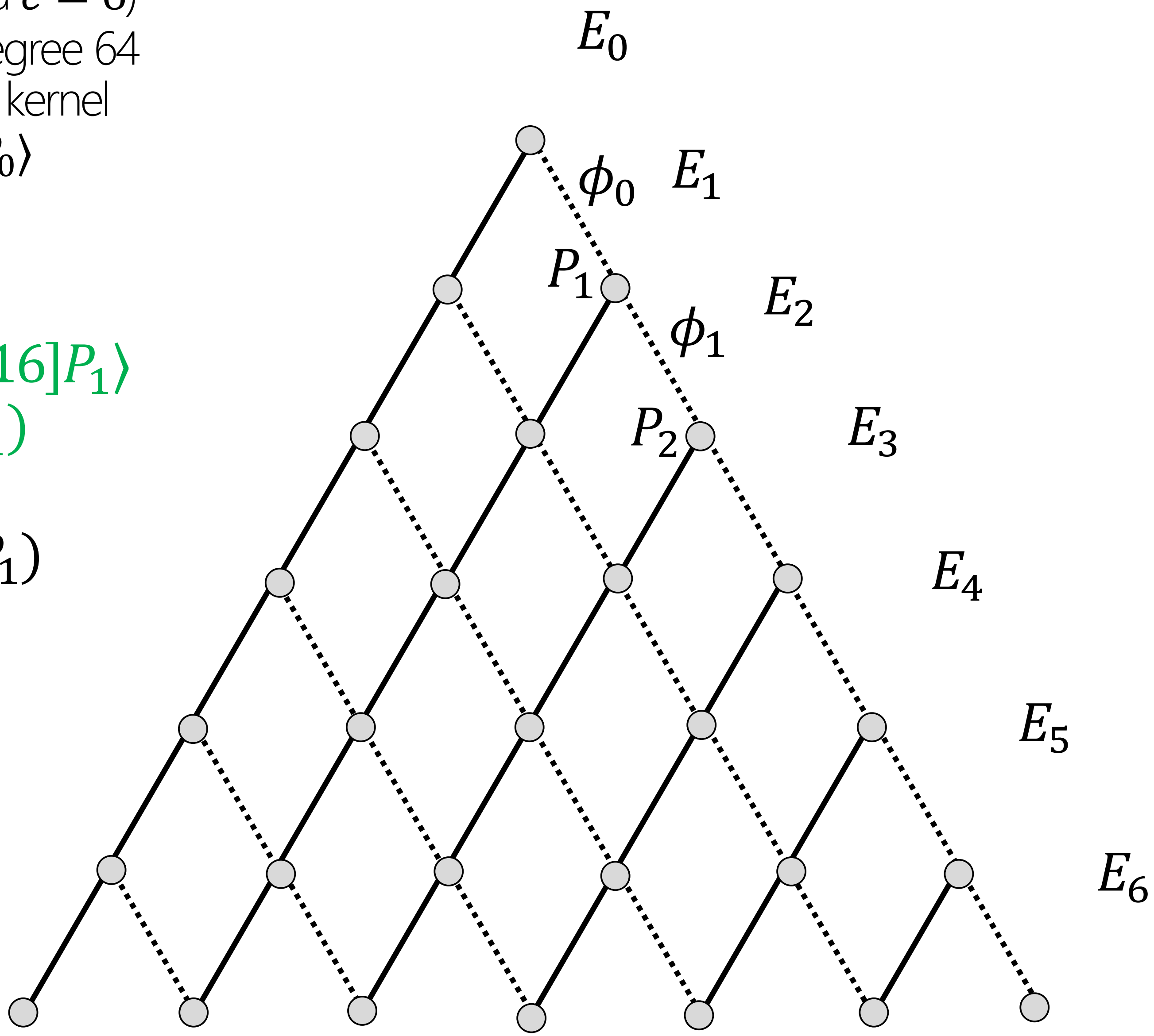


Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$$E_2 = E_1/\langle [16]P_1 \rangle = \phi_1(E_1)$$

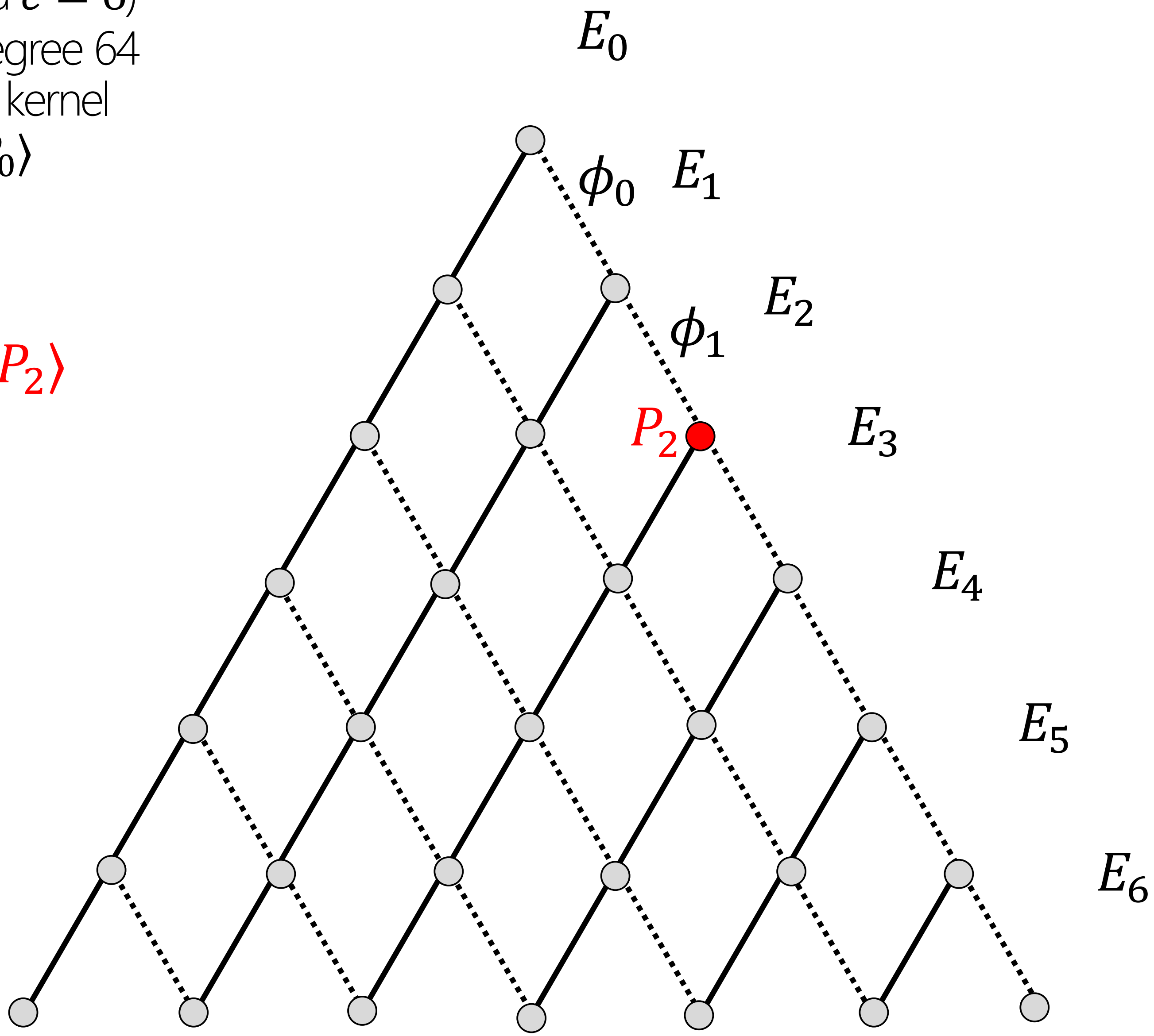
$$P_2 = \phi_1(P_1)$$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : \; E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

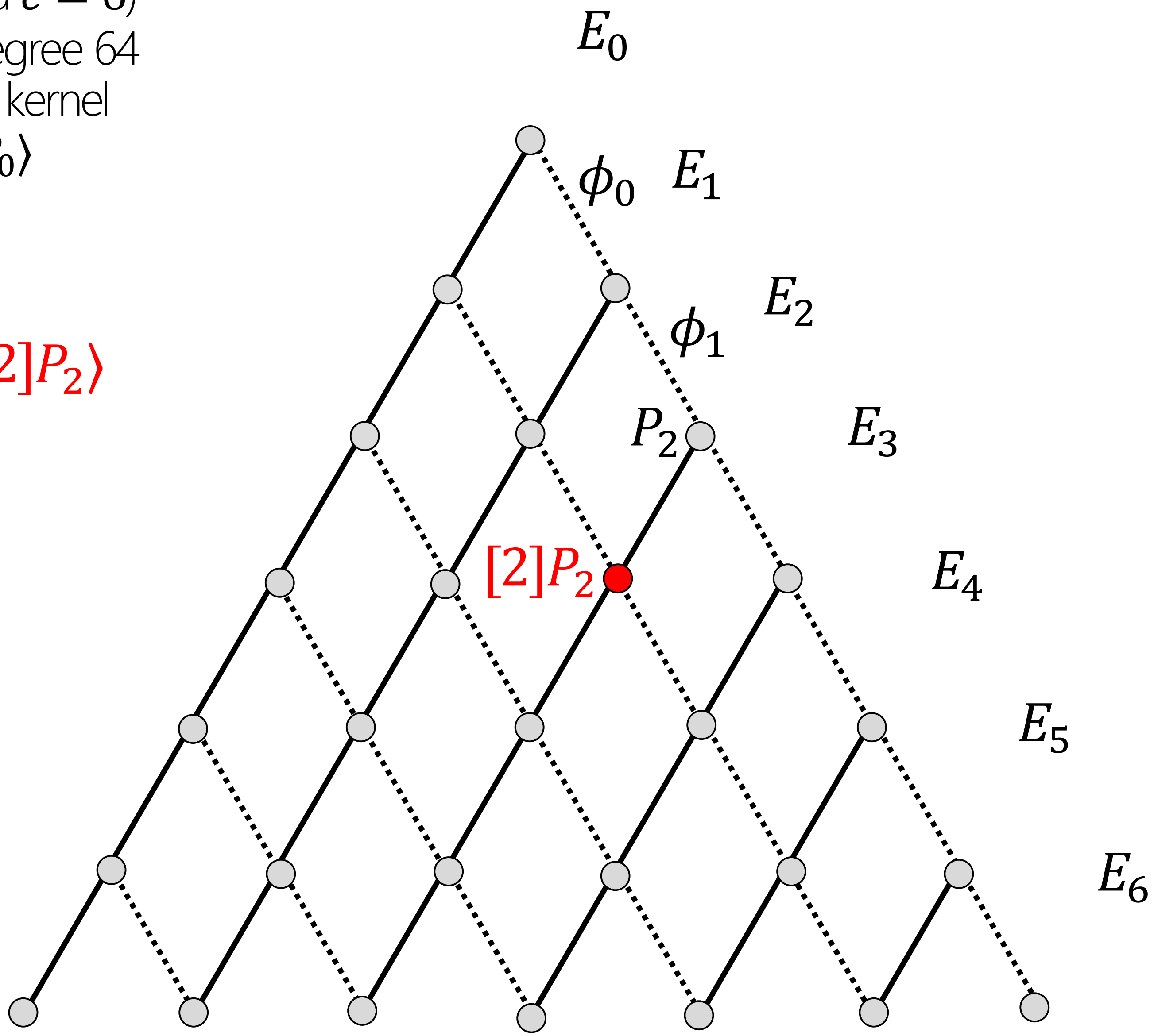
$E_6 = E_2 / \langle P_2 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

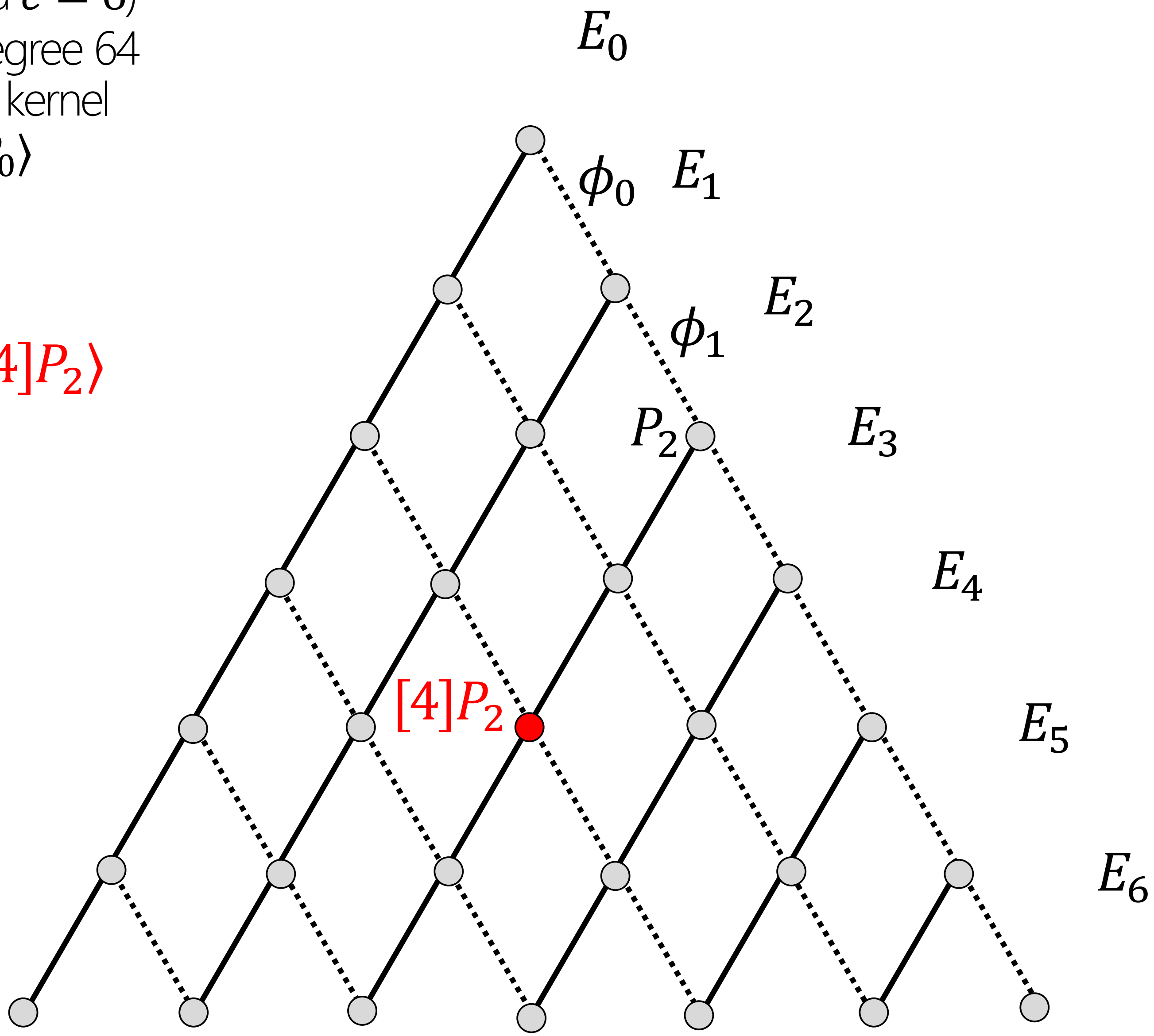
$E_5 = E_2/\langle [2]P_2 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : \ E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

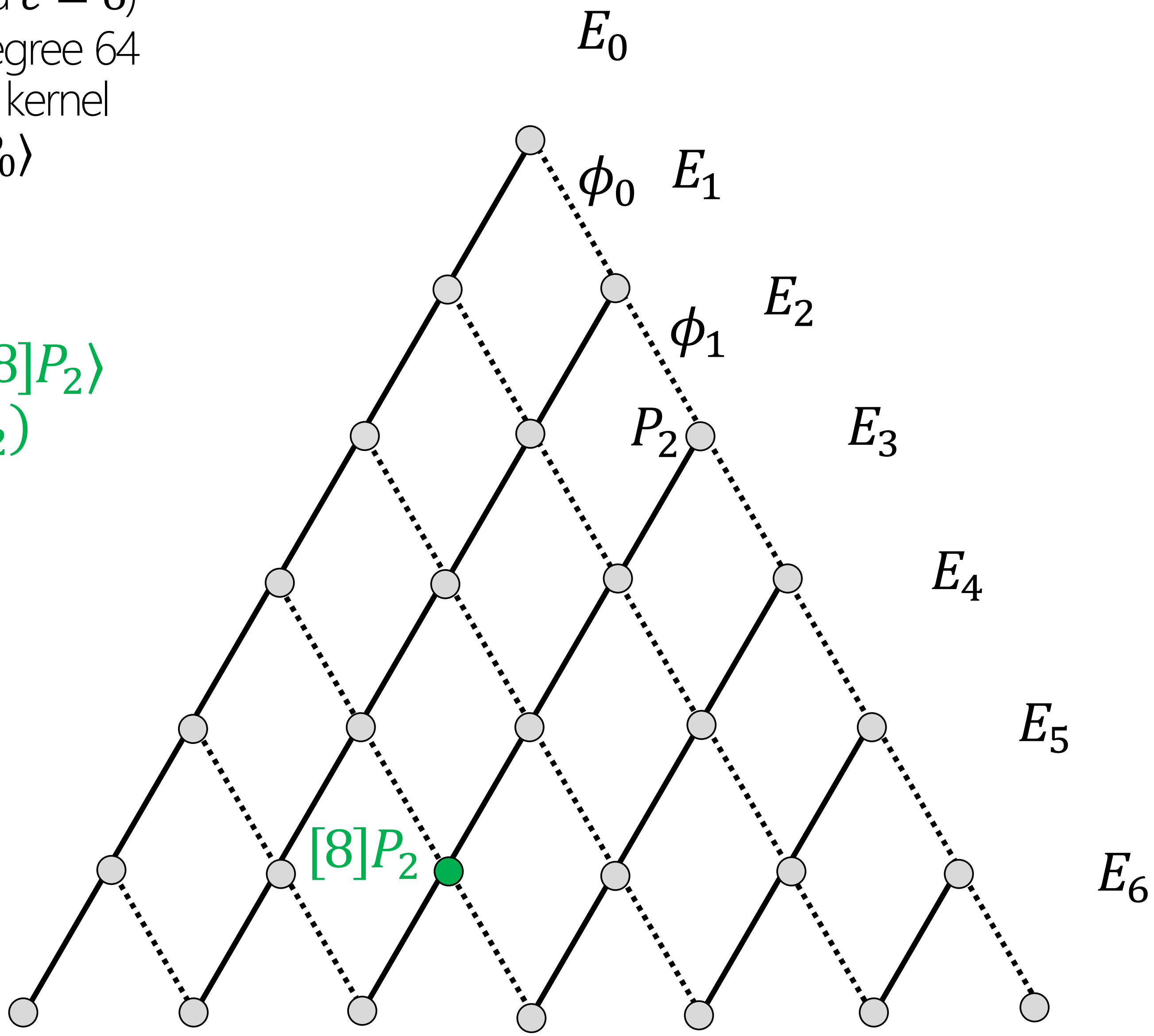
$E_4 = E_2/\langle [4]P_2 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : \ E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_3 = E_2/\langle [8]P_2 \rangle$
$= \phi_2(E_2)$

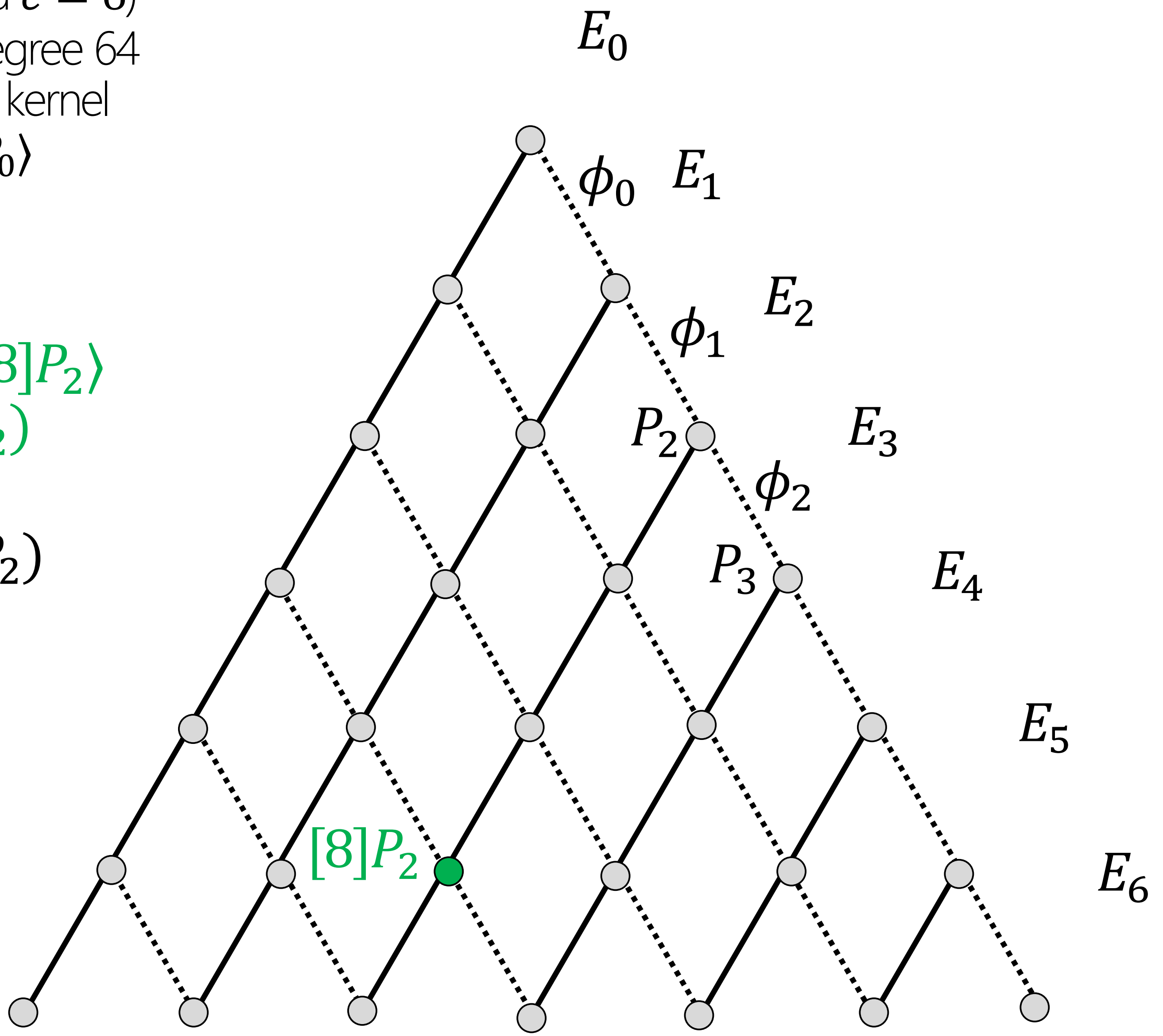


Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi :\; E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_3 = E_2/\langle [8]P_2 \rangle$
$= \phi_2(E_2)$

$P_3 = \phi_2(P_2)$



Computing ℓ^e degree isogenies

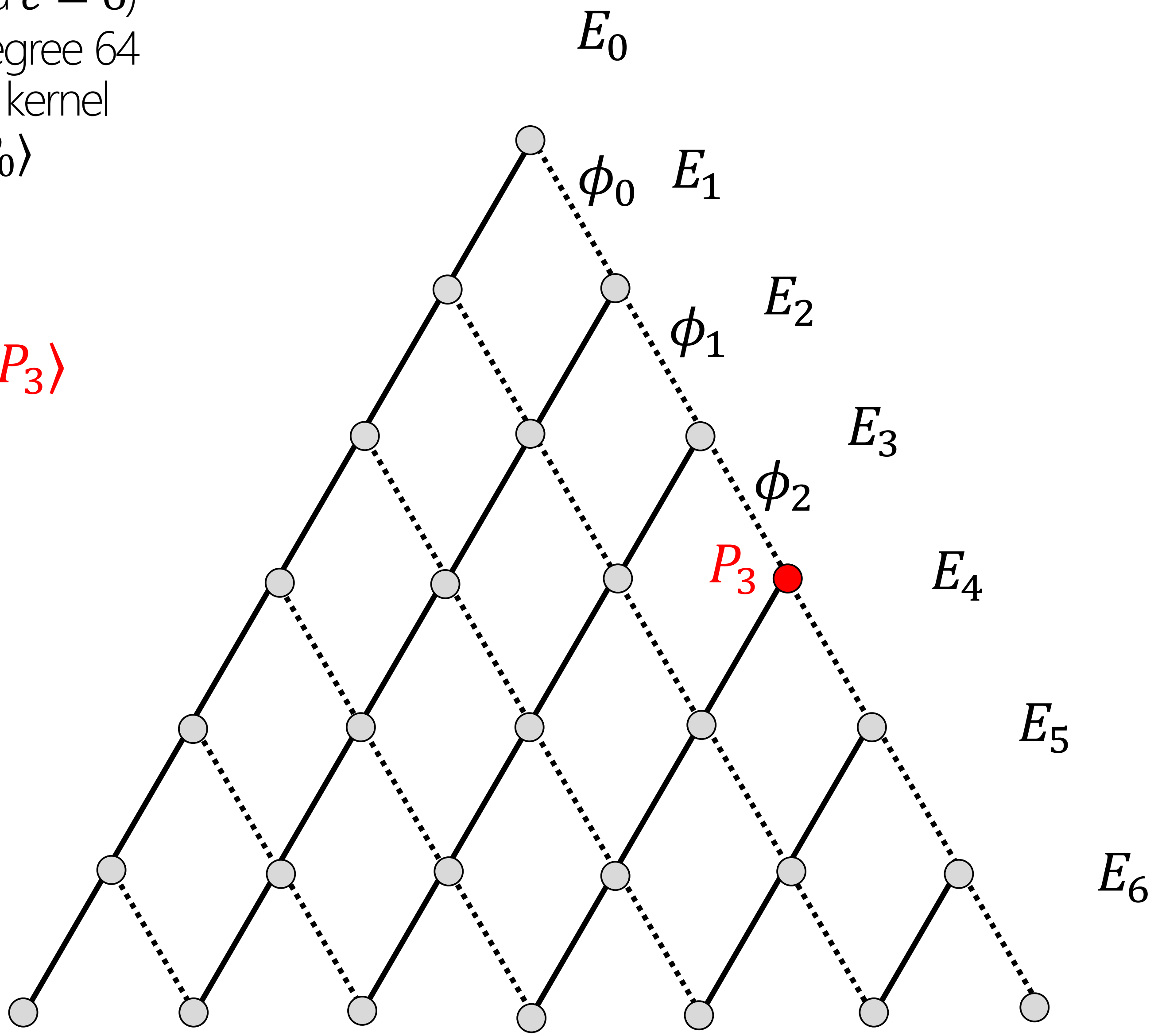
(suppose $\ell = 2$ and $e = 6$)

$\phi : \ E_0 \to E_6$ is degree 64

64 elements in its kernel

$\ker(\phi) = \langle P_0 \rangle$

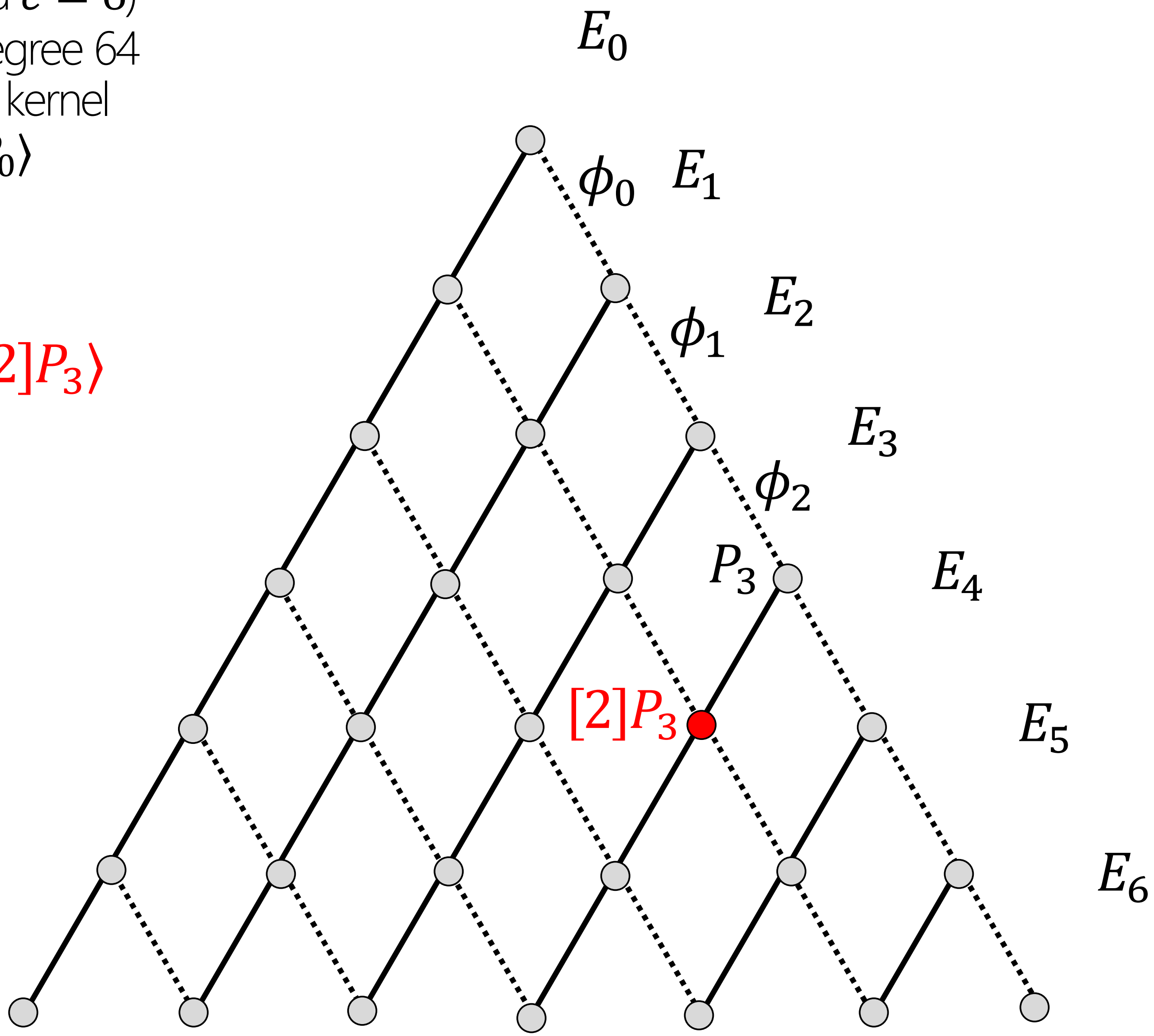
$E_6 = E_3 / \langle P_3 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

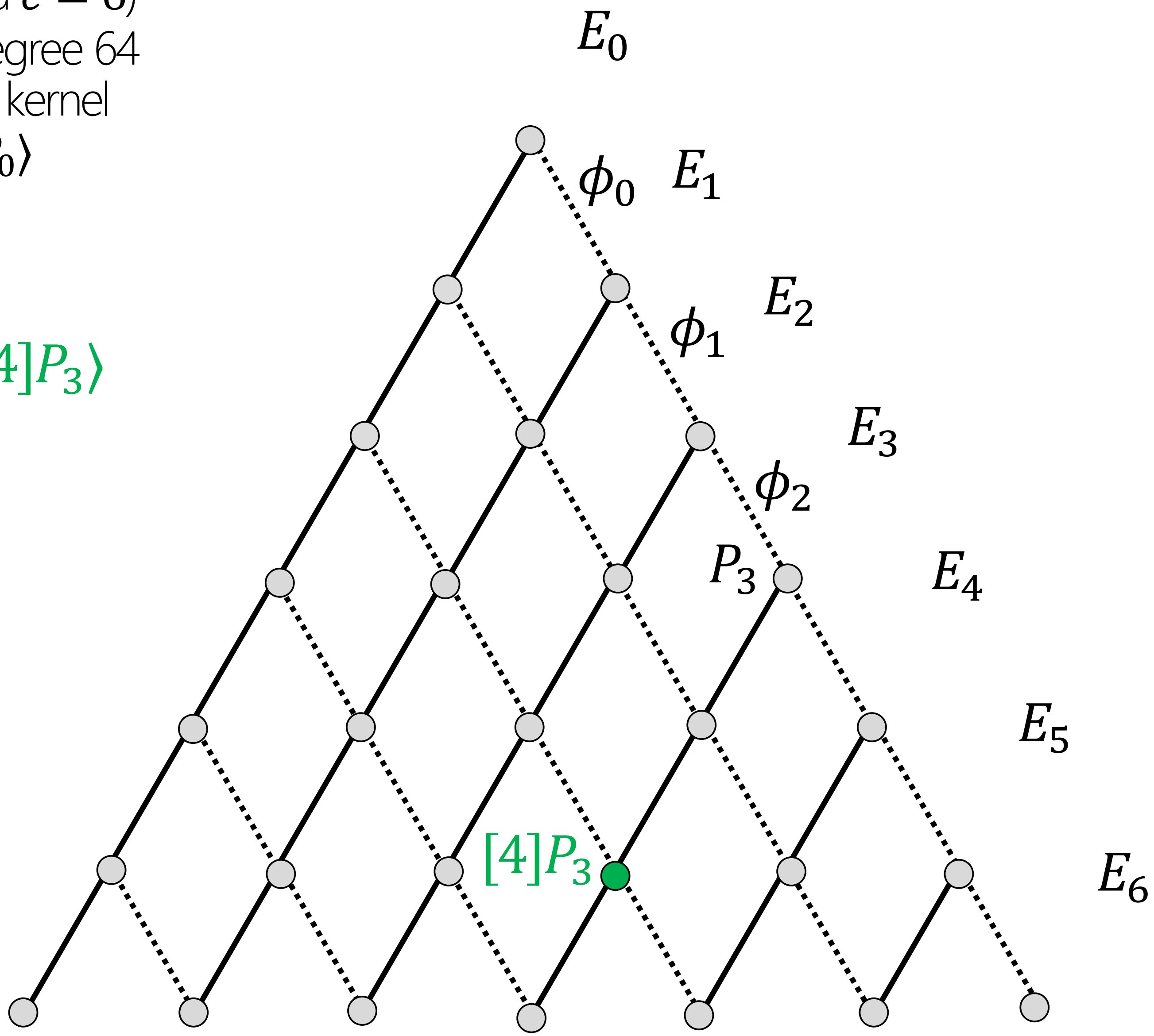
$E_5 = E_3/\langle [2]P_3 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi:\ E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_4 = E_3/\langle [4]P_3 \rangle$

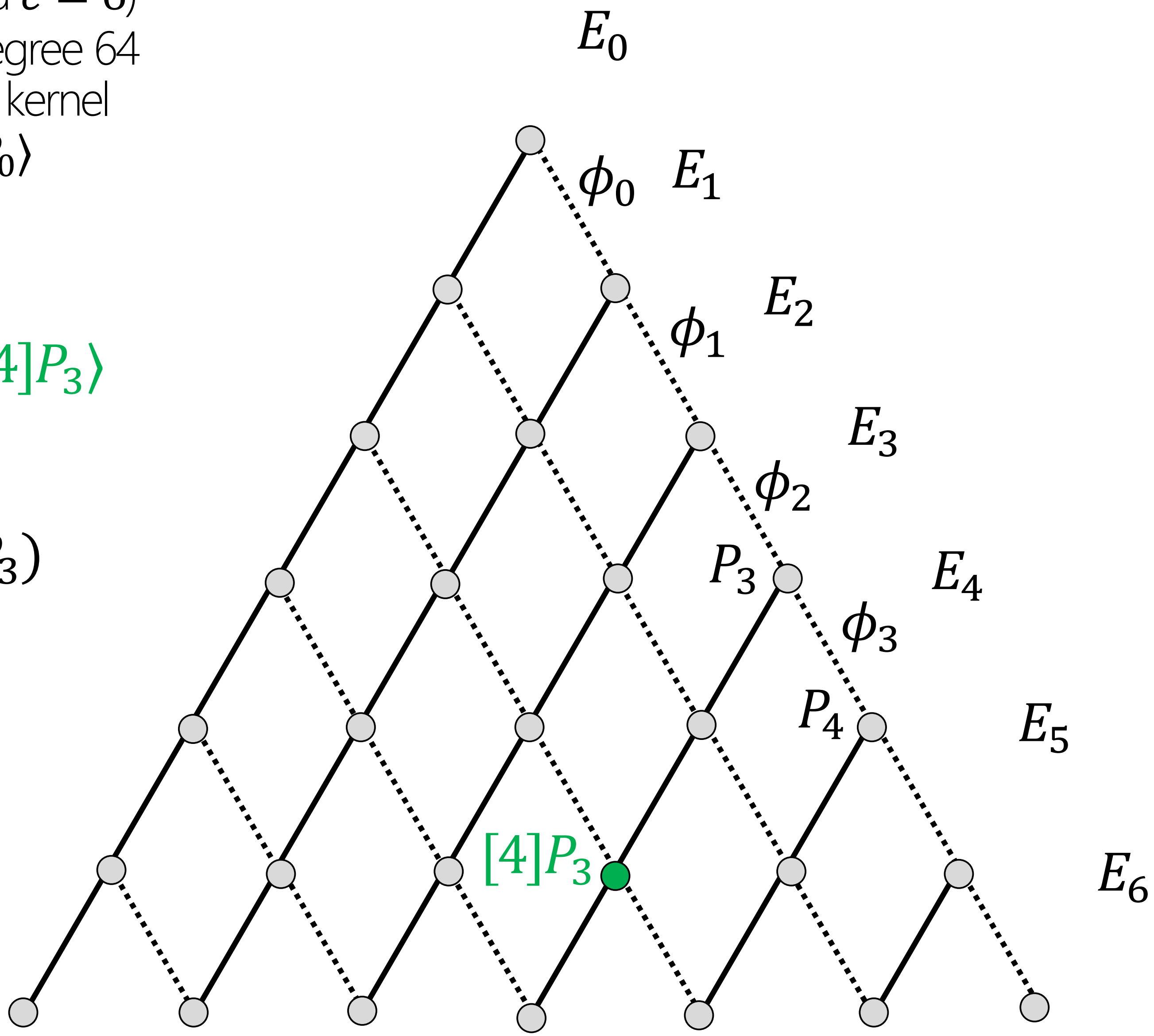


Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_4 = E_3/\langle [4]P_3 \rangle$

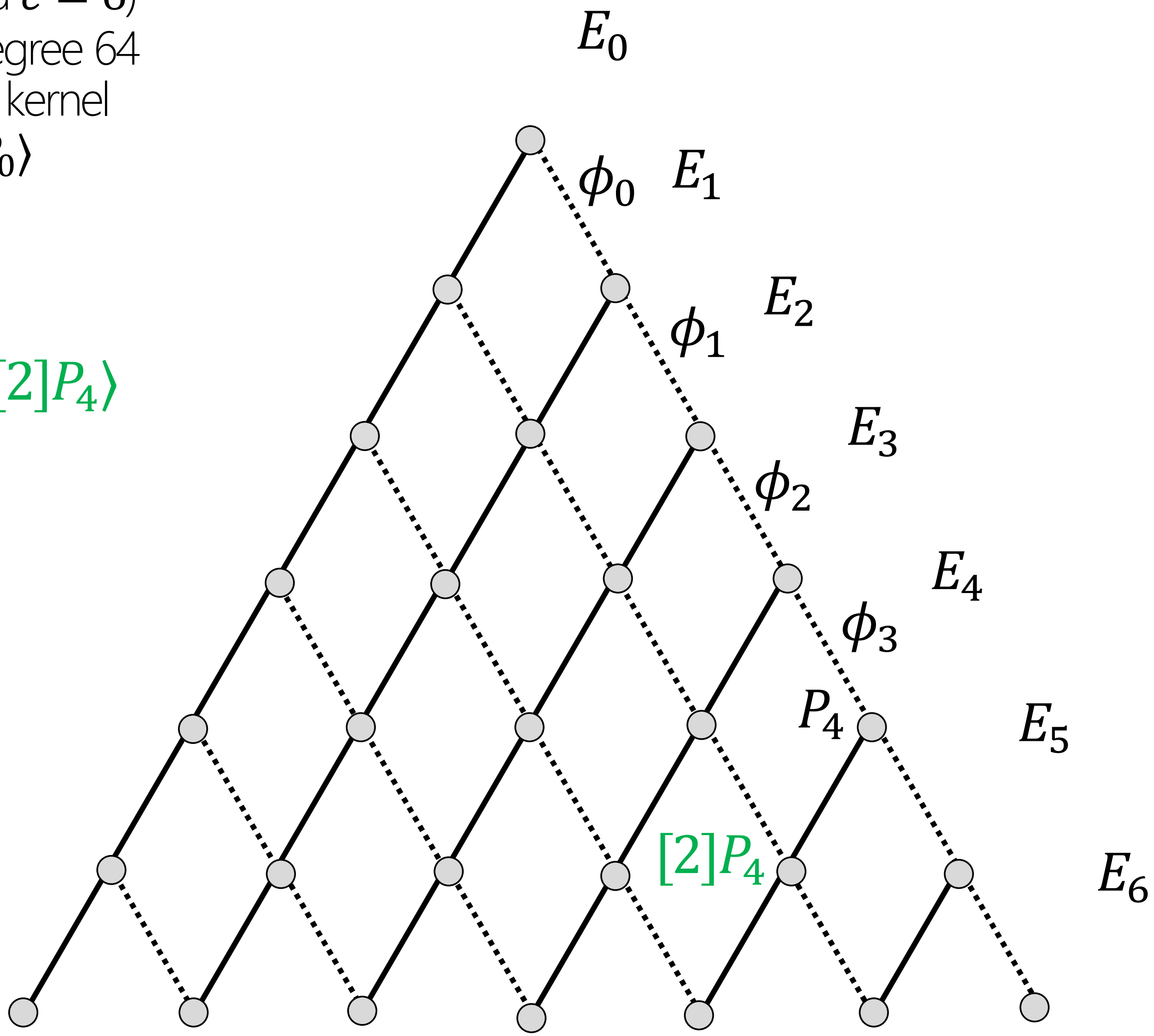
$P_4 = \phi_3(P_3)$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_5 = E_4 / \langle [2]P_4 \rangle$



Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)

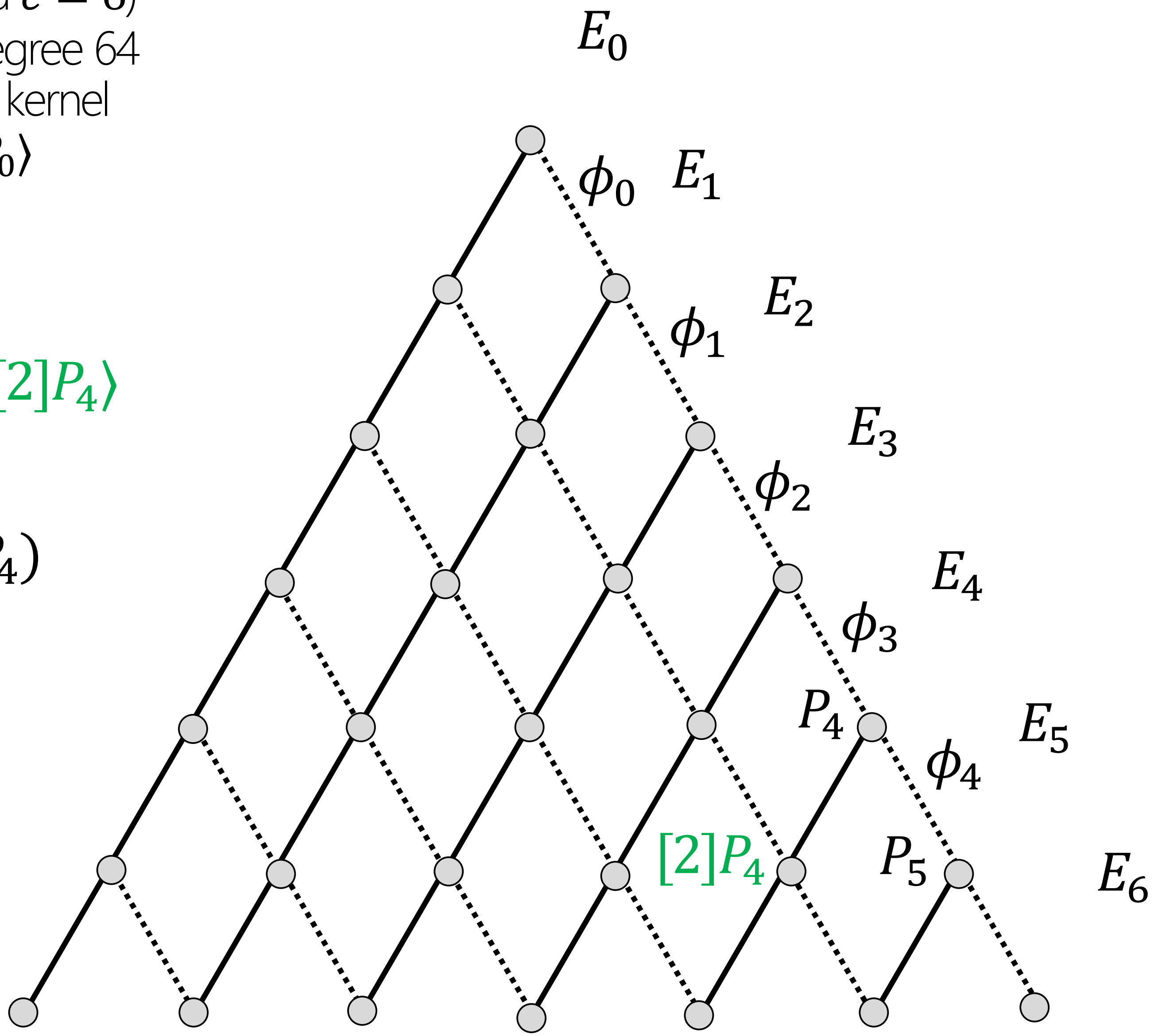
$\phi : \ E_0 \to E_6$ is degree 64

64 elements in its kernel

$\ker(\phi) = \langle P_0 \rangle$

$E_5 = E_4 / \langle [2] P_4 \rangle$

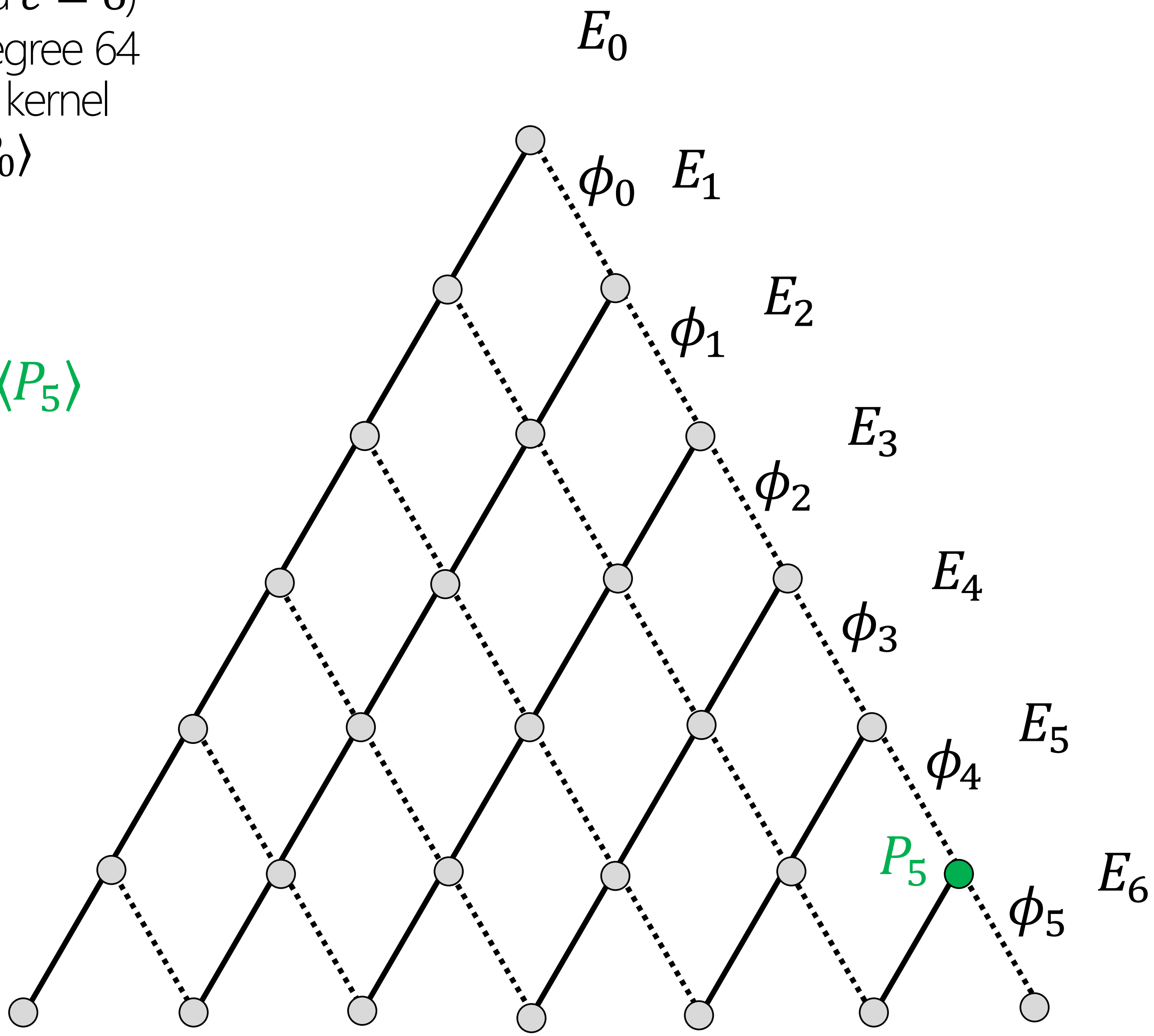
$P_5 = \phi_4(P_4)$



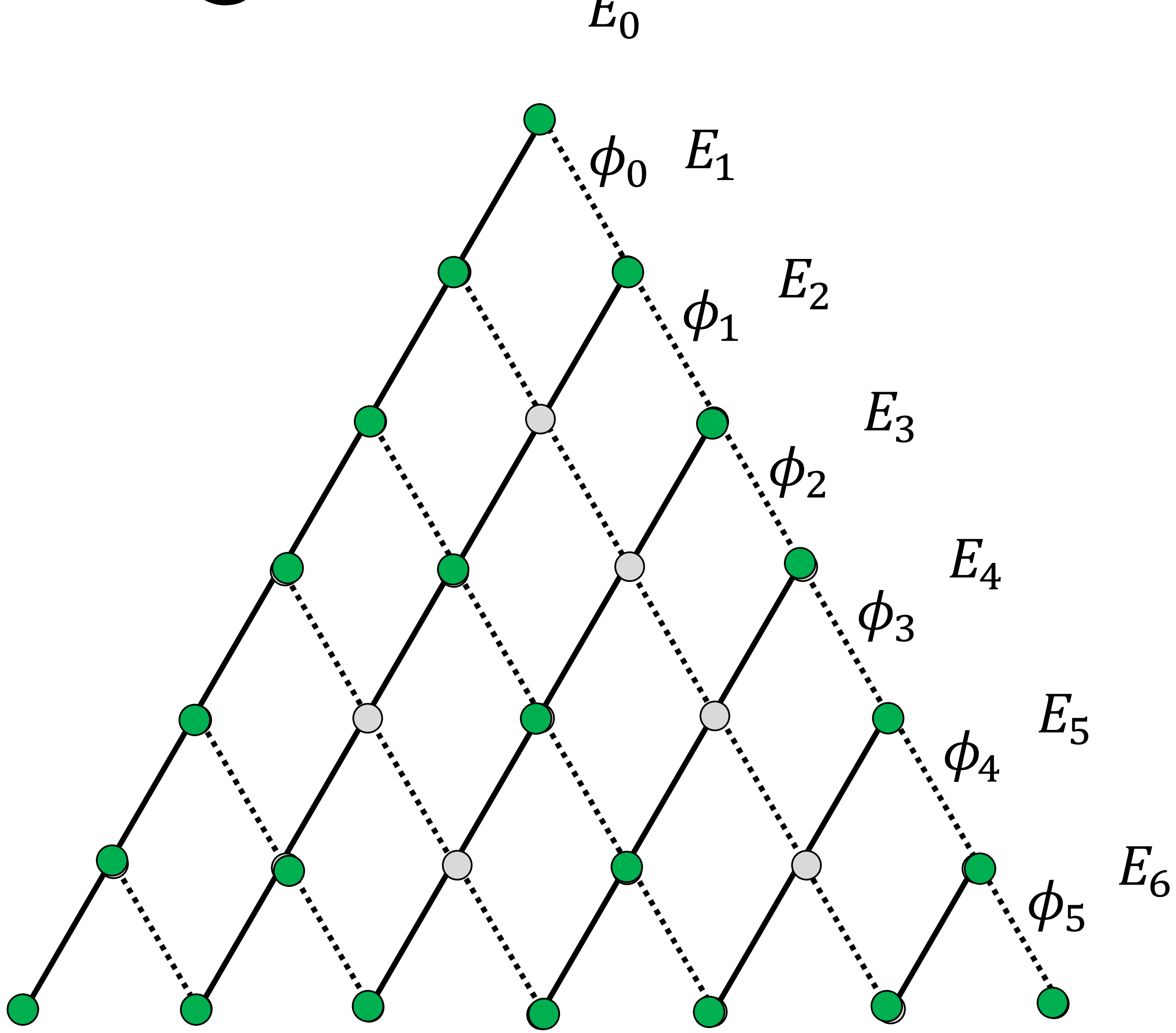
Computing ℓ^e degree isogenies

(suppose $\ell = 2$ and $e = 6$)
$\phi : E_0 \to E_6$ is degree 64
64 elements in its kernel
$\ker(\phi) = \langle P_0 \rangle$

$E_6 = E_5/\langle P_5 \rangle$

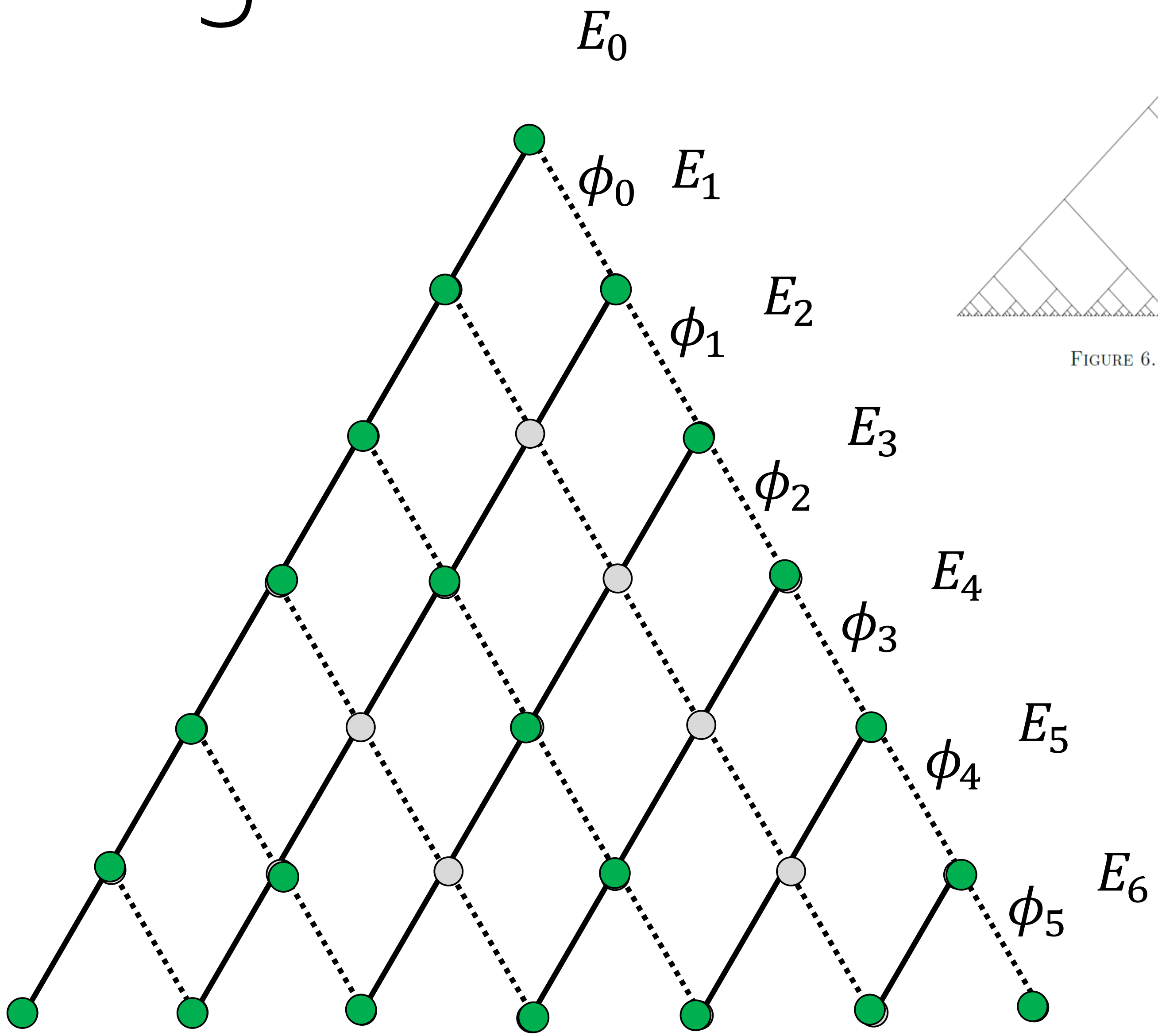


Optimal strategies



Optimal strategies

$$n^2 \rightarrow n \log n$$



TOWARDS QUANTUM-RESISTANT CRYPTOSYSTEMS FROM ISOGENIES

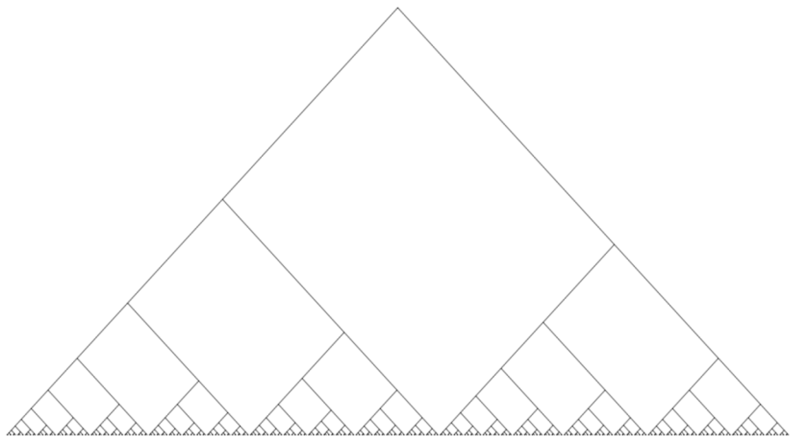
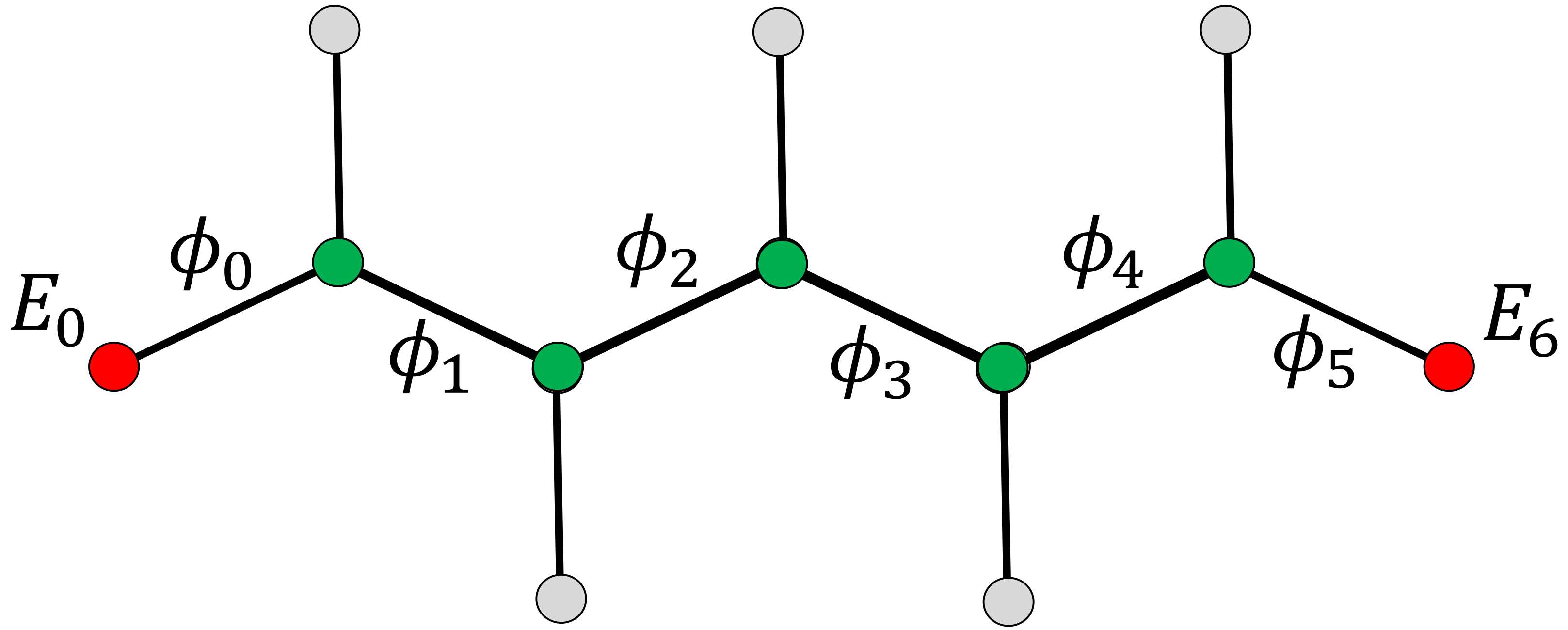


FIGURE 6. Optimal strategy for $n = 512, p = 4.6, q = 2.8$.

Computing ℓ^e degree isogenies

$$\phi\ :\ E_0 \to E_6$$

$$\phi = \phi_5 \circ \phi_4 \circ \phi_3 \circ \phi_2 \circ \phi_1 \circ \phi_0$$



Rest of talk: given $\boldsymbol{E}, \boldsymbol{E'}$, find path (of known length)…

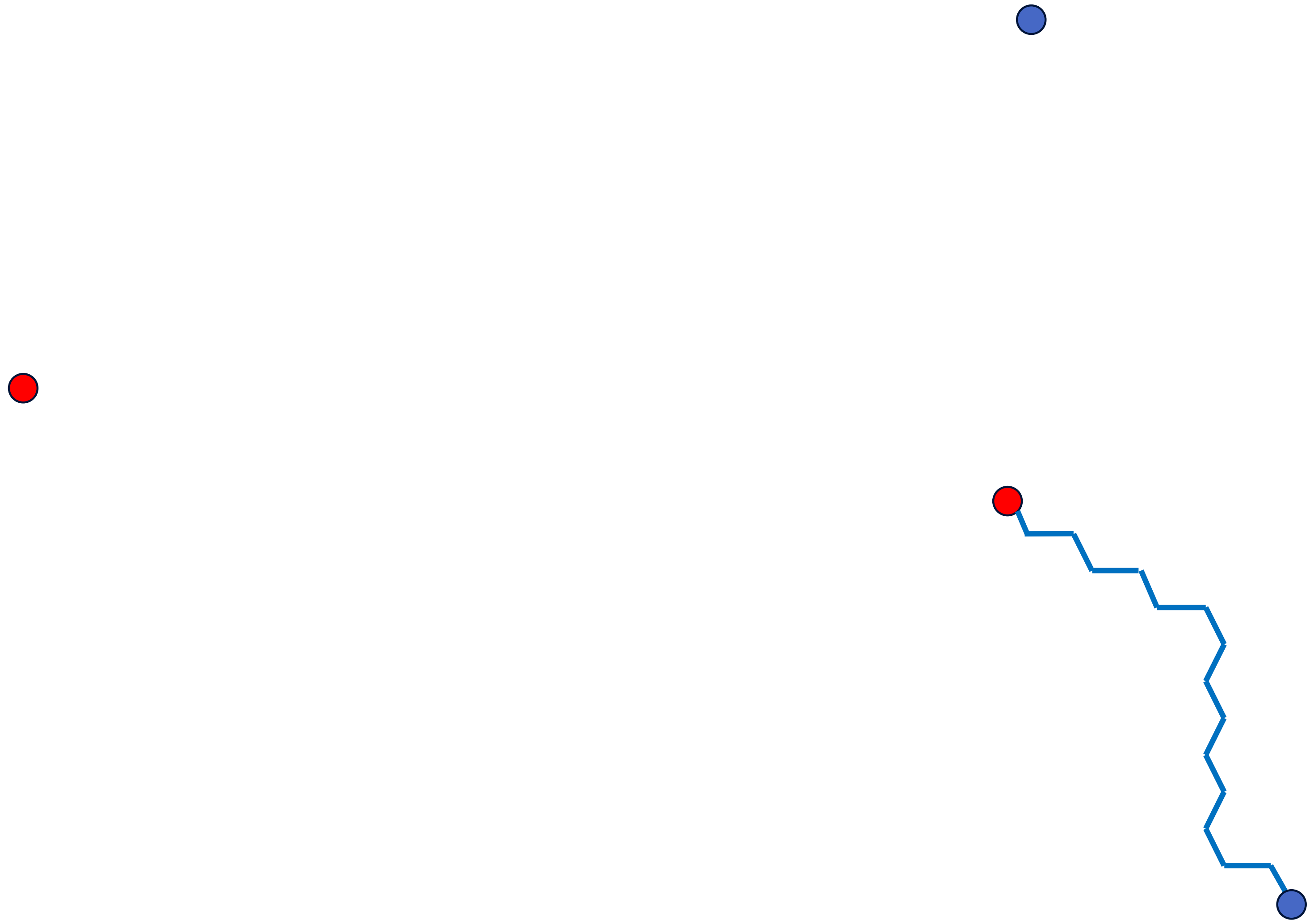
Claw algorithm: meet-in-the-middle

Given E and $E' = \phi(E)$, with ϕ degree ℓ^e, find ϕ

Claw algorithm: meet-in-the-middle

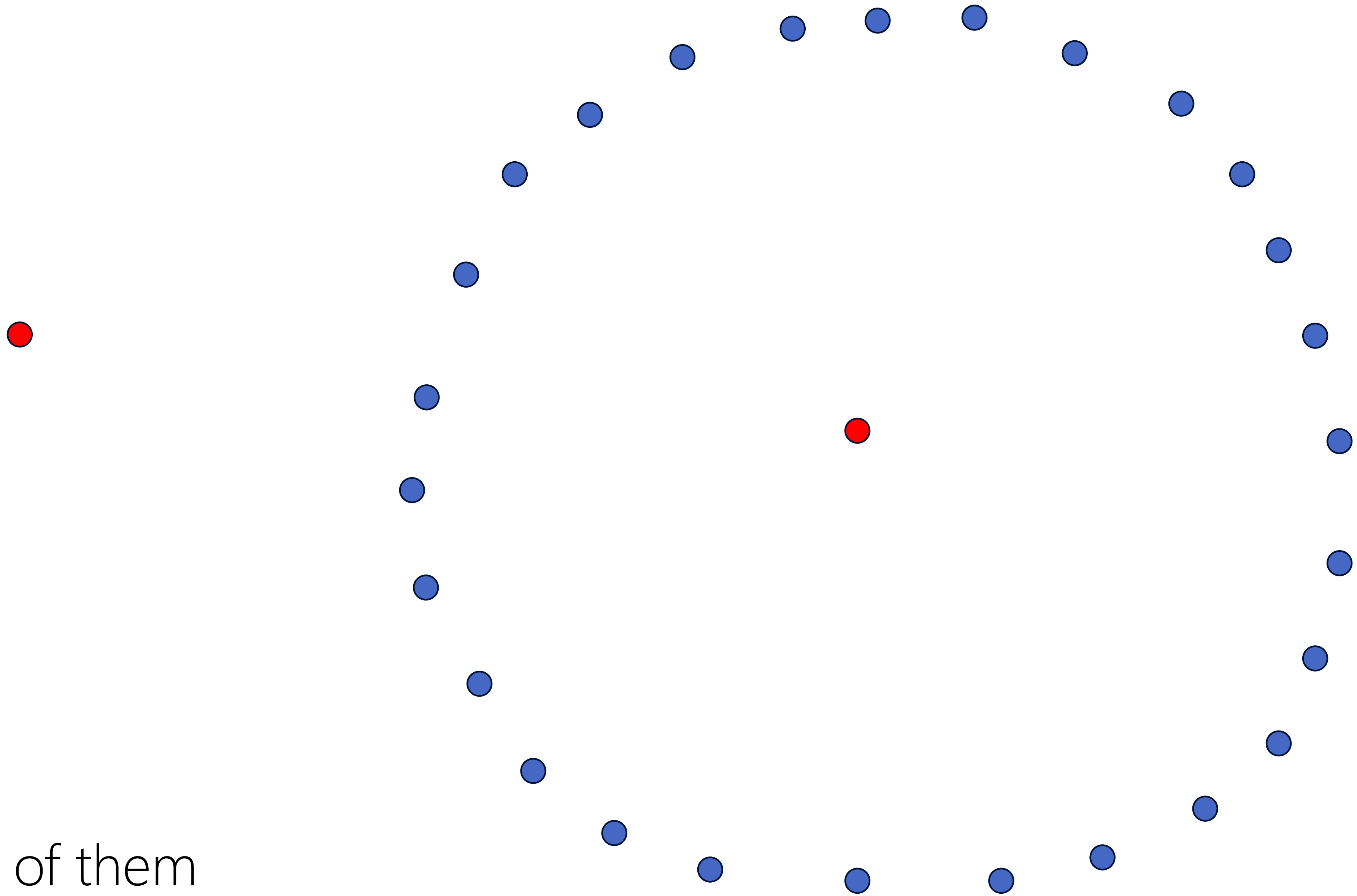
Compute and store $\ell^{e/2}$-isogenies on one side

Claw algorithm: meet-in-the-middle



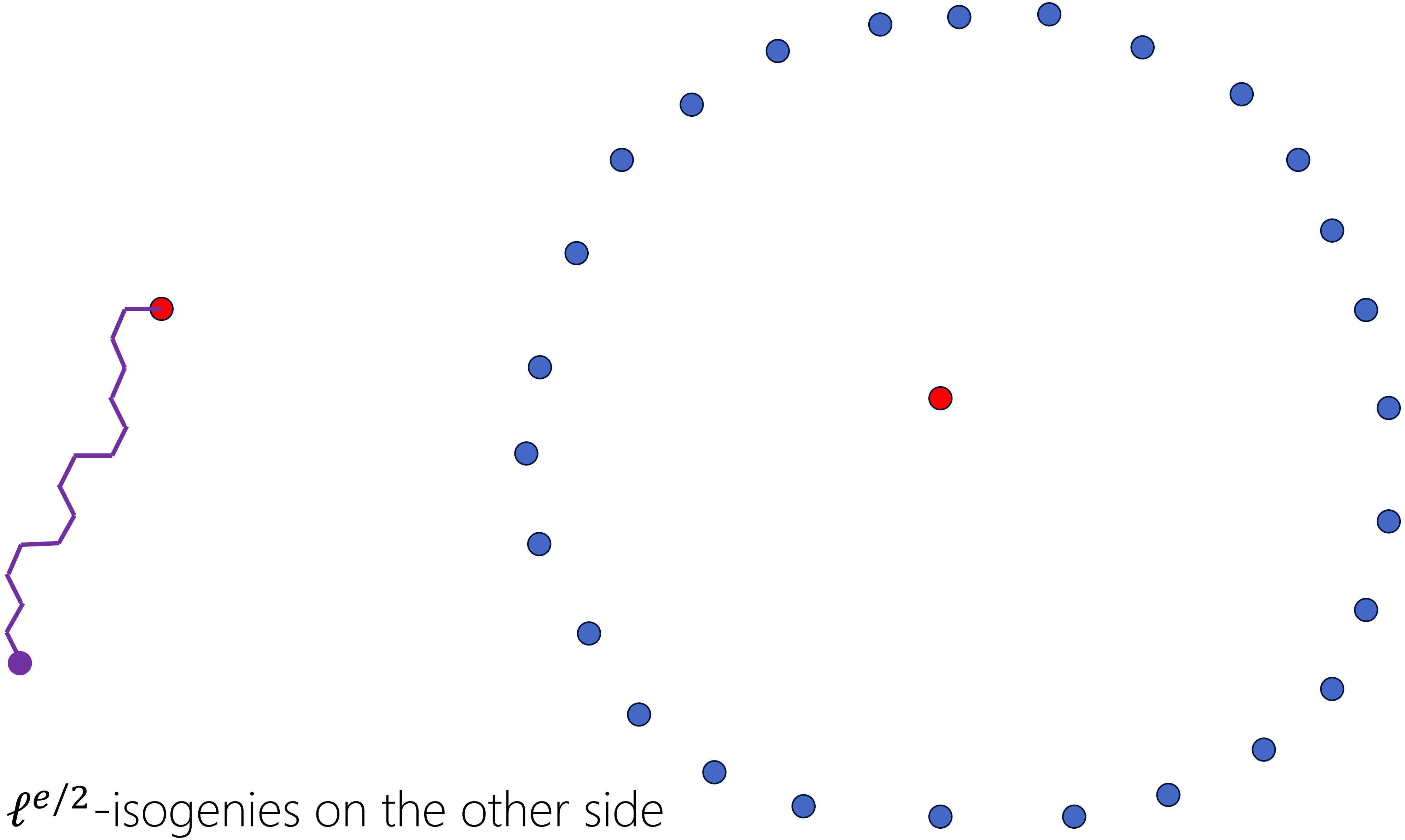
Compute and store $\ell^{e/2}$-isogenies on one side

Claw algorithm: meet-in-the-middle



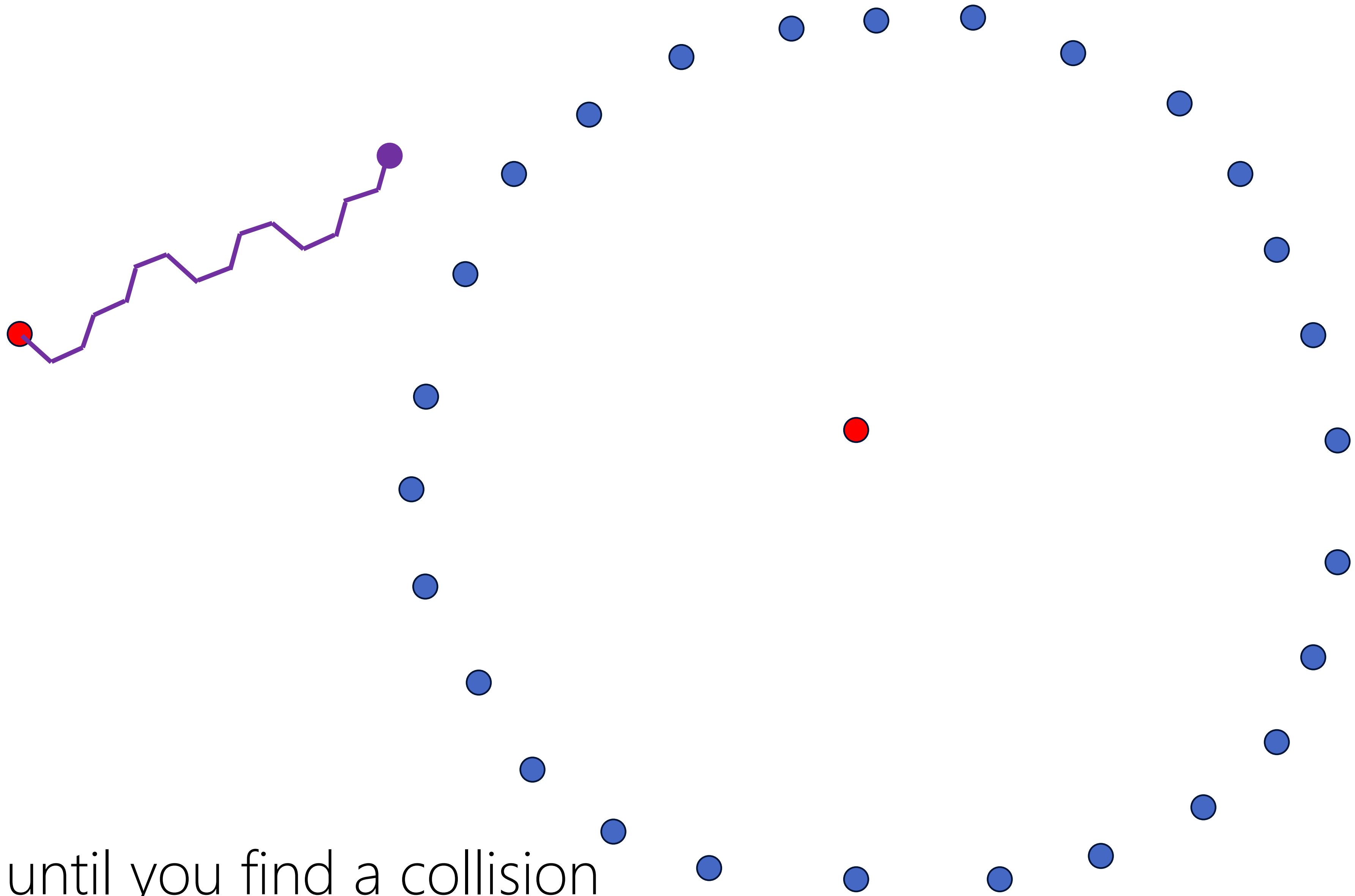
… until you have all of them

Claw algorithm: meet-in-the-middle



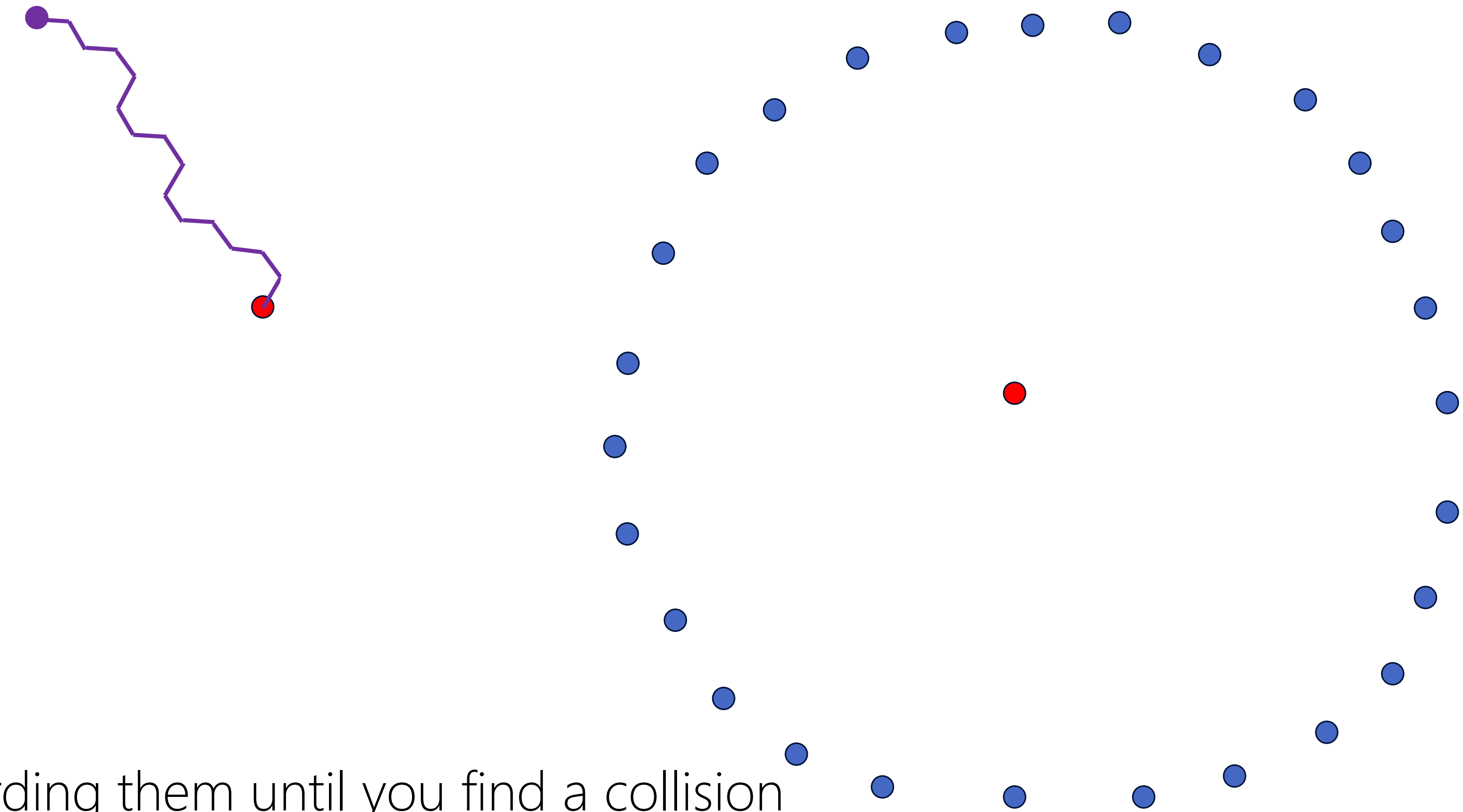
Now compute $\ell^{e/2}$-isogenies on the other side

Claw algorithm: meet-in-the-middle



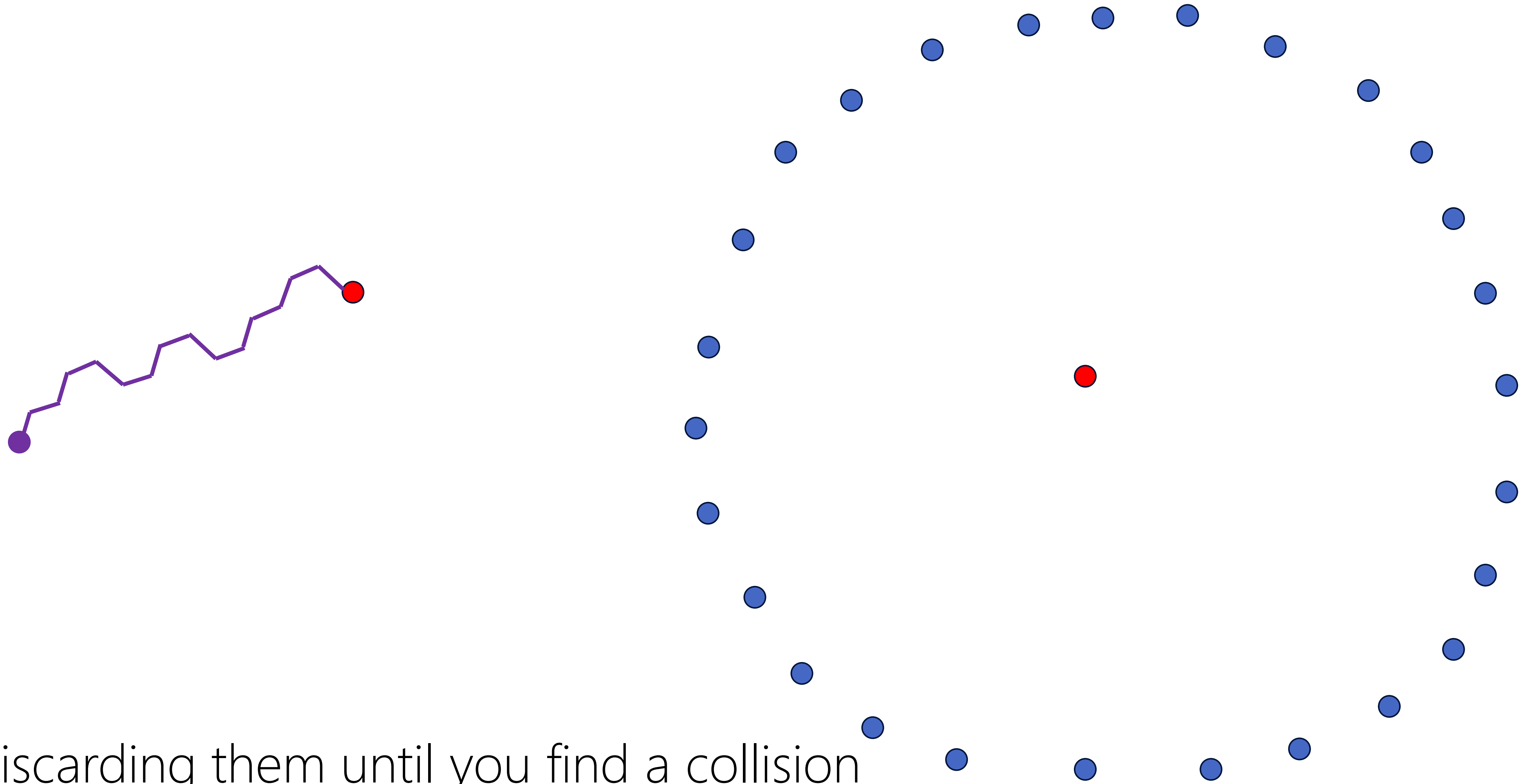
… discarding them until you find a collision

Claw algorithm: meet-in-the-middle



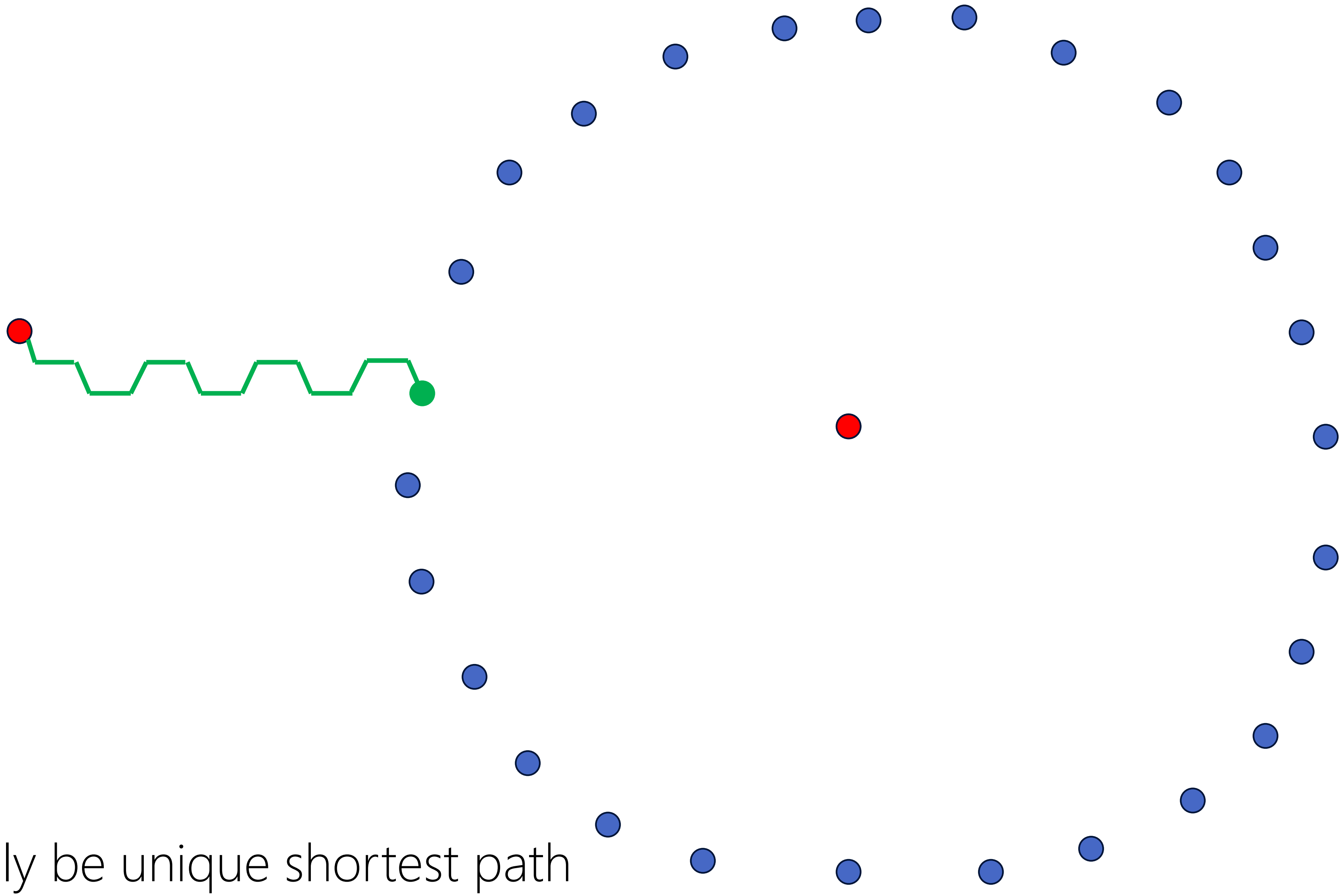
… discarding them until you find a collision

Claw algorithm: meet-in-the-middle



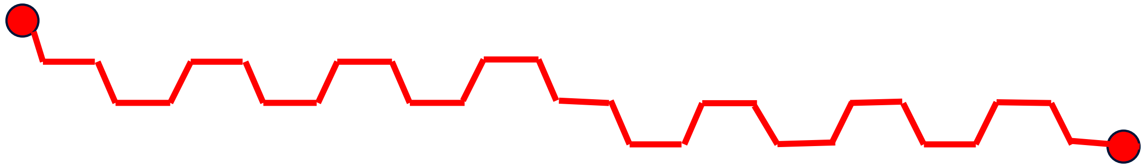
… discarding them until you find a collision

Claw algorithm: meet-in-the-middle



Collision will most likely be unique shortest path

Claw algorithm: meet-in-the-middle



This path describes secret isogeny $\phi : E \to E'$

Claw algorithm: classical analysis

- There are $O(\ell^{e/2})$ curves $\ell^{e/2}$-isogenous to E' (the blue nodes)

  thus $O(\ell^{e/2}) = O(p^{1/4})$ classical memory

- There are $O(\ell^{e/2})$ curves $\ell^{e/2}$-isogenous to E' (the blue nodes), and there are $O(\ell^{e/2})$ curves $\ell^{e/2}$-isogenous to E (the purple nodes)

  thus $O(\ell^{e/2}) = O(p^{1/4})$ classical time

- **Best (known) attacks:** classical $O(p^{1/4})$ and quantum $O(p^{1/6})$
- **Confidence:** both complexities are optimal for a black-box claw attack

The curves and their security estimates

$$p = 2^{e_A}3^{e_B} - 1$$

Target Security Level	**Name** (SIKEp+ $\lceil \log_2 p \rceil$)	$(\boldsymbol{e_A}, \boldsymbol{e_B})$	$\boldsymbol{k}$	$\mathbf{2^{k-1}}$	**min** $(\sqrt{\mathbf{2}^{e_A}}, \sqrt{\mathbf{3}^{e_3}})$	$\sqrt{\mathbf{2}^{k}}$	**min** $(\sqrt[3]{\mathbf{2}^{e_2}}, \sqrt[3]{\mathbf{3}^{e_3}})$
NIST 1	SIKEp503	$(250,159)$	128	2^{127}	2^{125}	2^{64}	2^{83}
NIST 3	SIKEp761	$(372,239)$	192	2^{191}	2^{186}	2^{96}	2^{124}
NIST 5	SIKEp964	$(486,301)$	256	2^{255}	2^{238}	2^{128}	2^{159}

classical quantum

Since submission…

cryptanalysis

- Adj, Cervantes-Vázquez, Chi-Domínguez, Menezes, Rodríguez-Henríquez: *On the cost of computing isogenies between supersingular elliptic curves* (ia.cr/2018/313)

- Jaques-Schanck: *Quantum cryptanalysis in the RAM model: claw-finding attacks on SIKE* (ia.cr/2019/103)

- C-Longa-Naehrig-Renes-Virdia: *Improved classical cryptanalysis of the computational supersingular isogeny problem* (ia.cr/2019/XXX)

compression

- Zanon, Simplicio Jr, Pereira, Doliskani, Barreto: *Faster key compression for isogeny-based cryptosystems* (ia.cr/2017/1143)

Jaques-Schanck (ia.cr/2019/103)

- Models allow direct classical-quantum comparison: best known quantum algorithms do not achieve significant advantage over classical
- (w.r.t. Tani and Grover) In certain attack scenarios classical security is the limiting factor for achieving a specified security level
- *"Our conclusion is that an adversary with enough memory to run Tani's algorithm with the query-optimal parameters could break SIKE faster by using the classical control hardware to run vOW"*

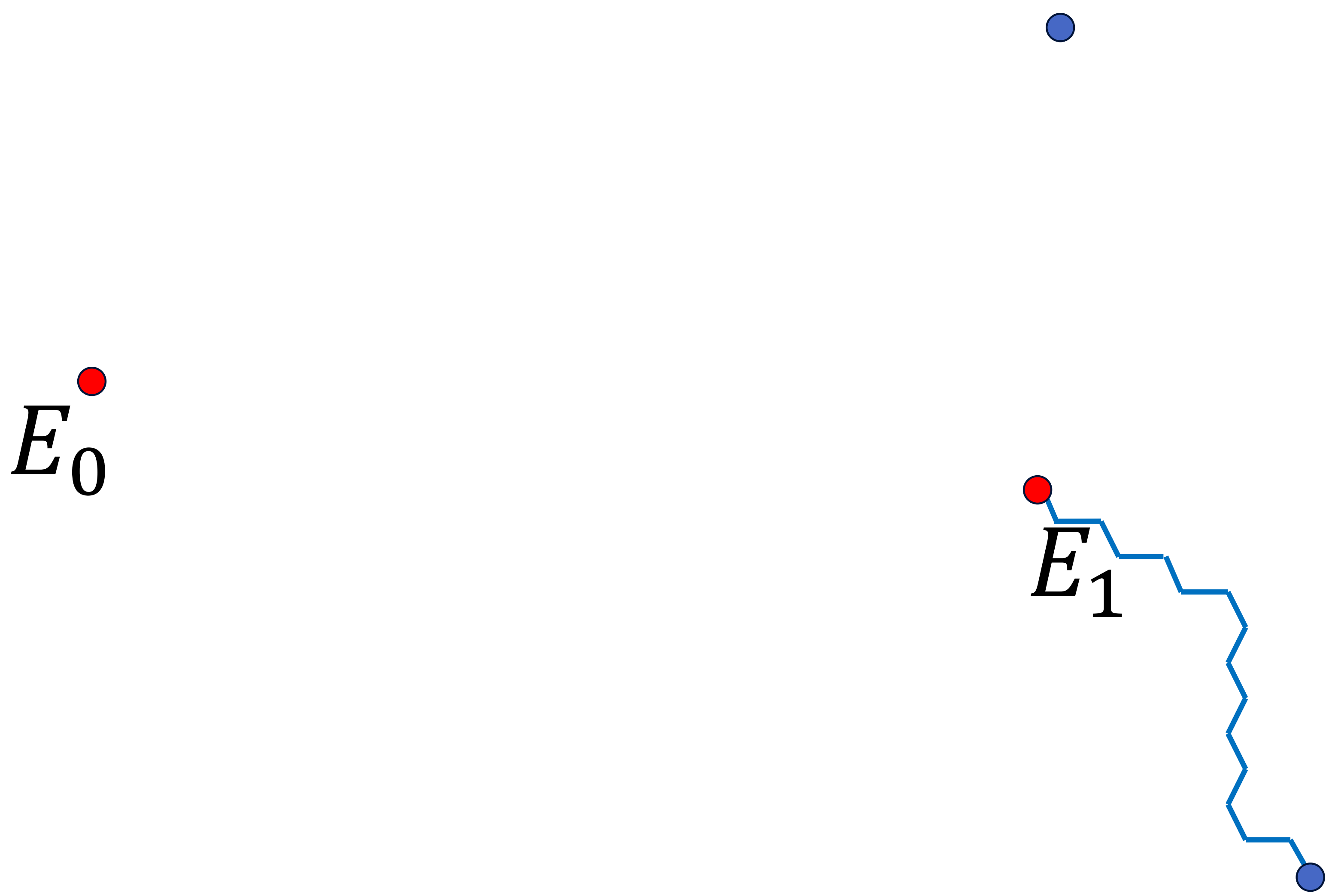
van Oorschot-Wiener

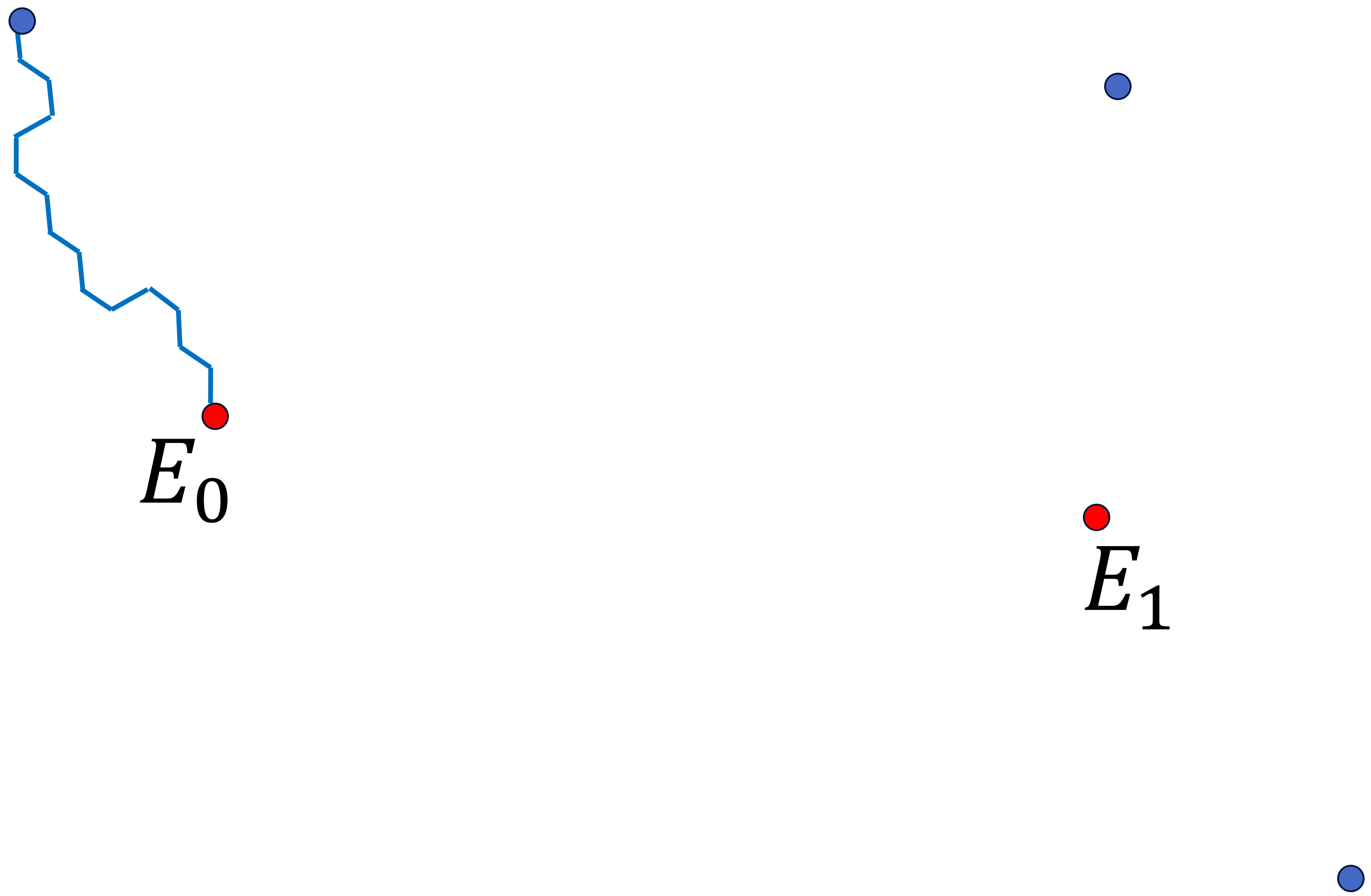
Do not have enough memory to MitM, so run a deterministic function that combines both sides into a set S

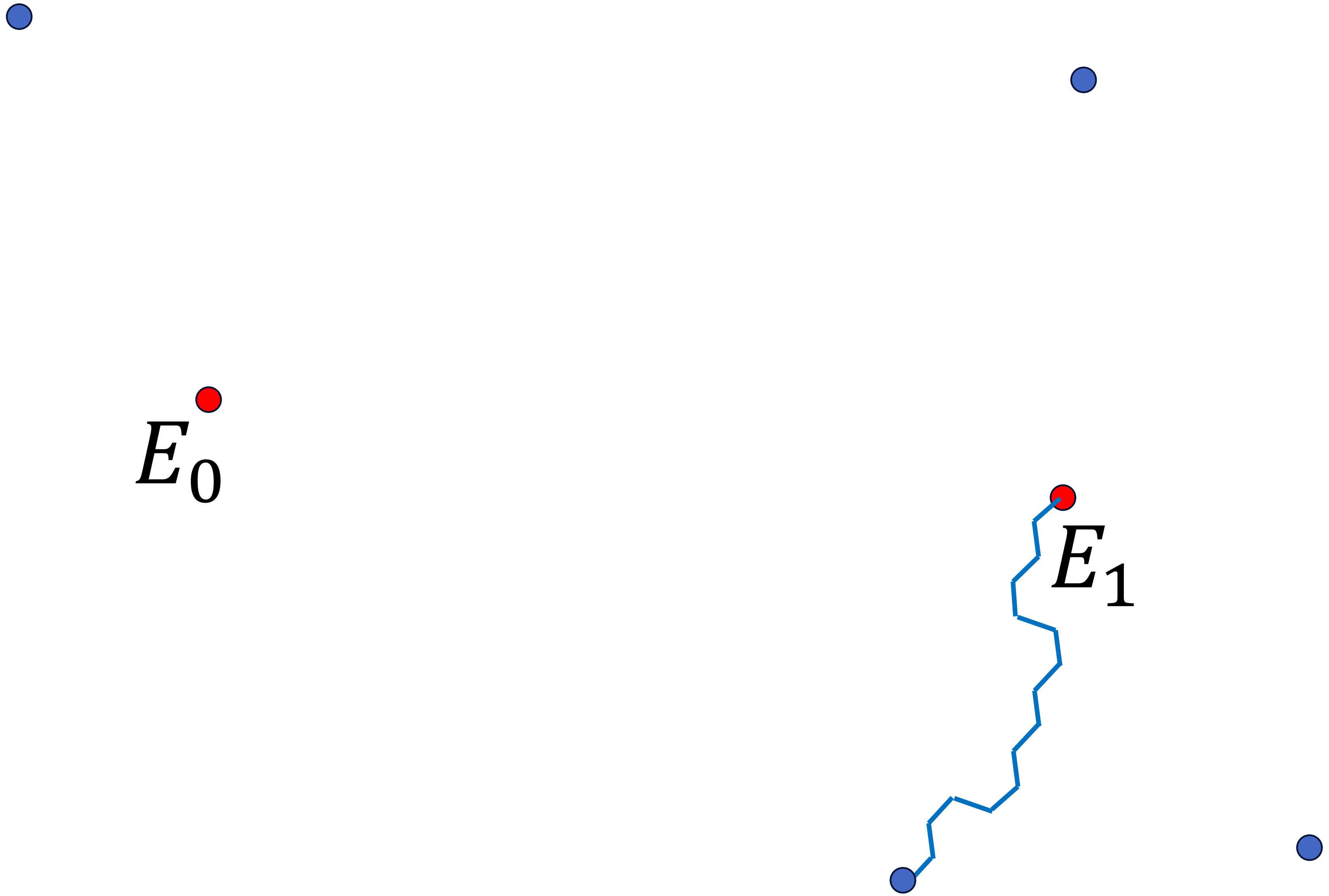
$$f_n: \quad S \to S$$
$$x_i \mapsto x_{i+1}$$

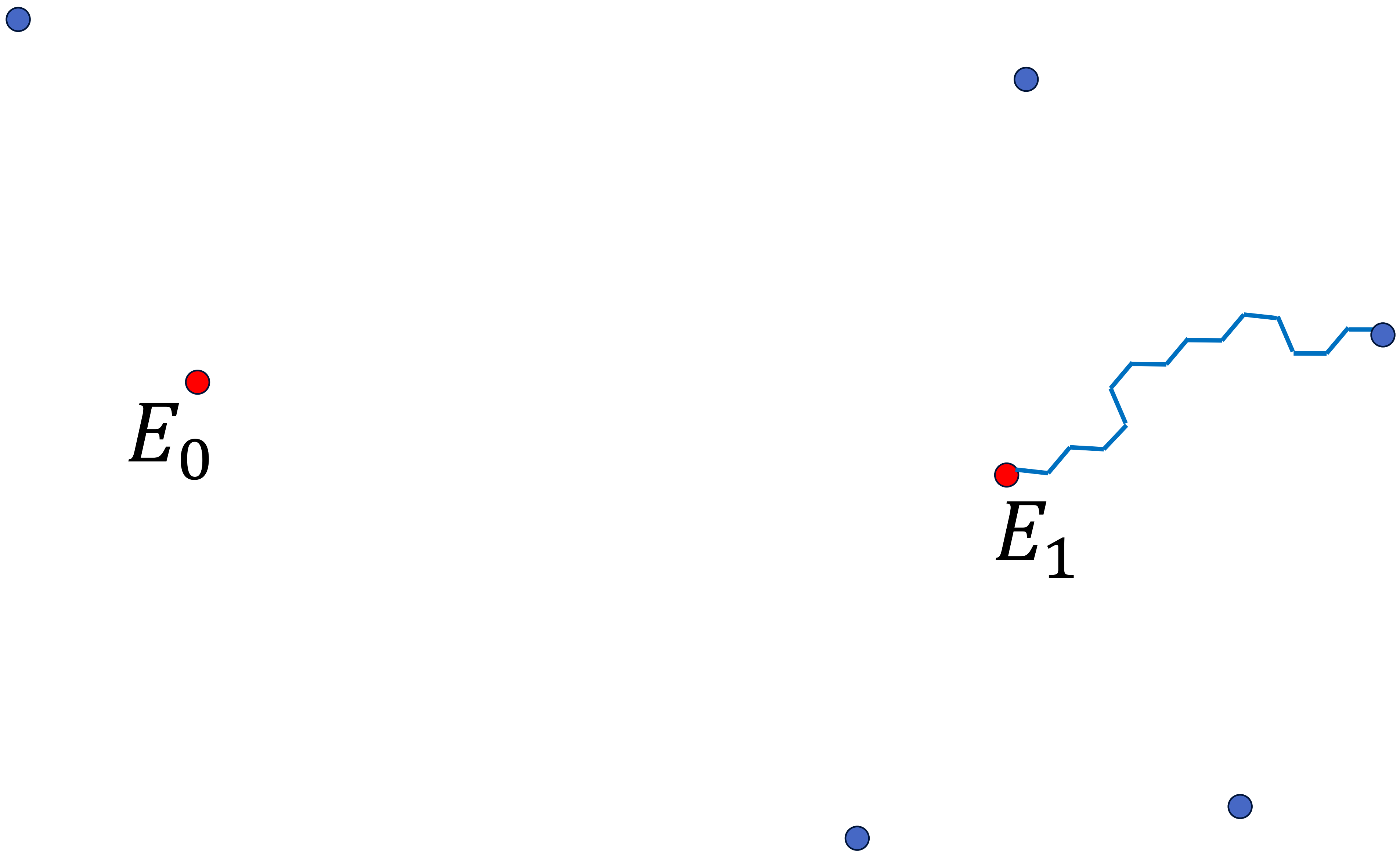
f_n : a half-sized isogeny + ϵ

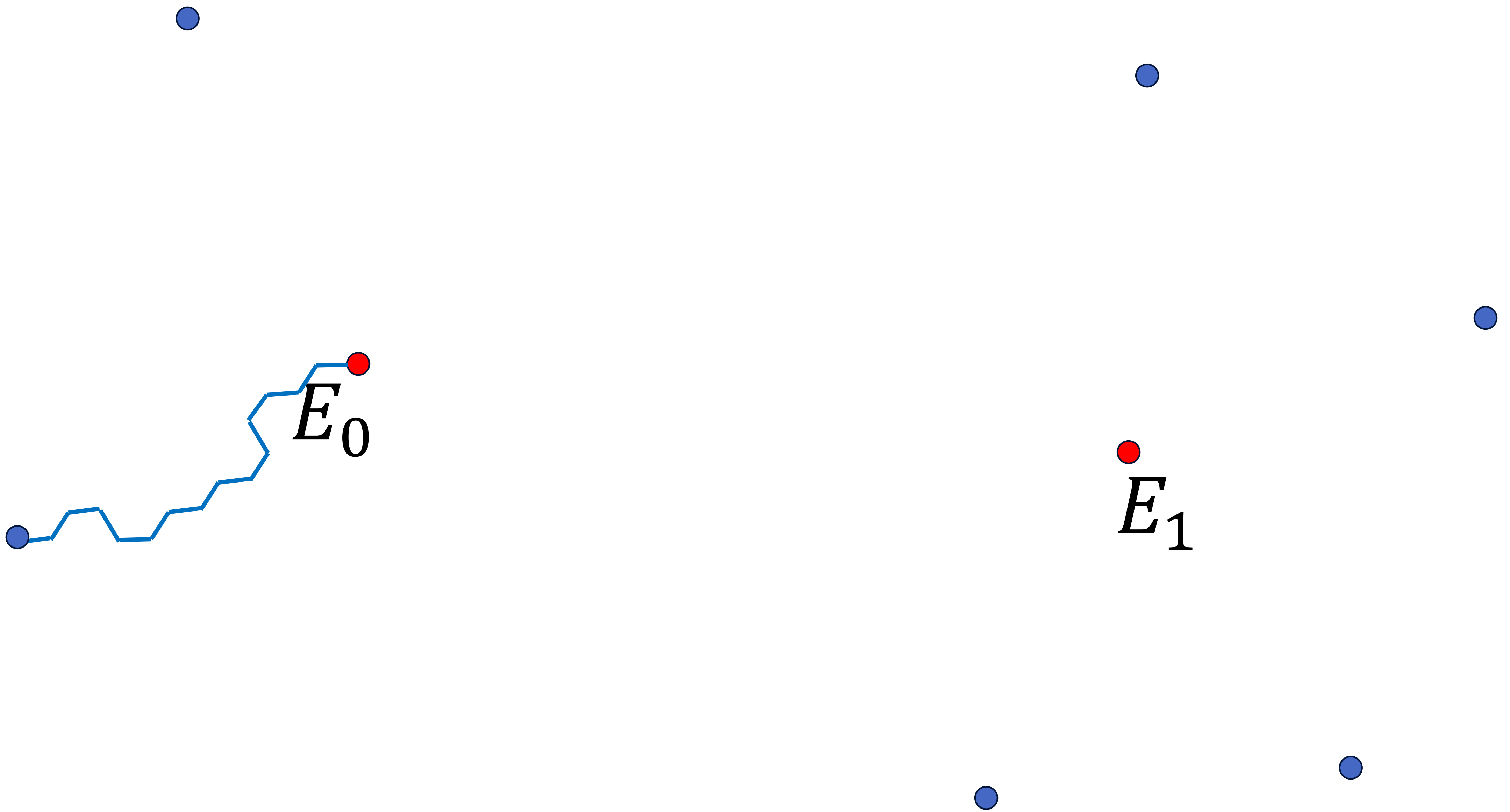

E_0
E_1

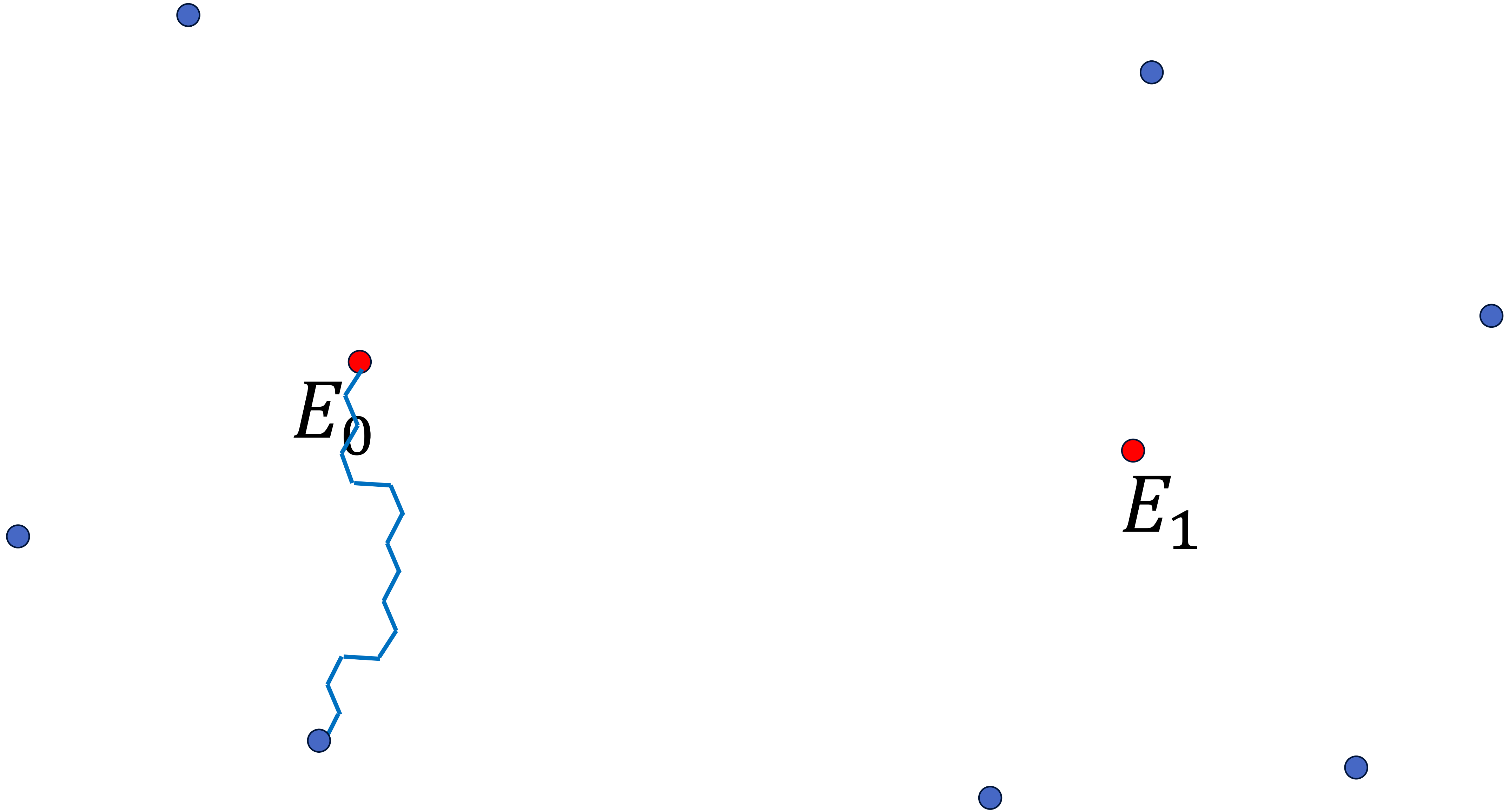

E_0
E_1

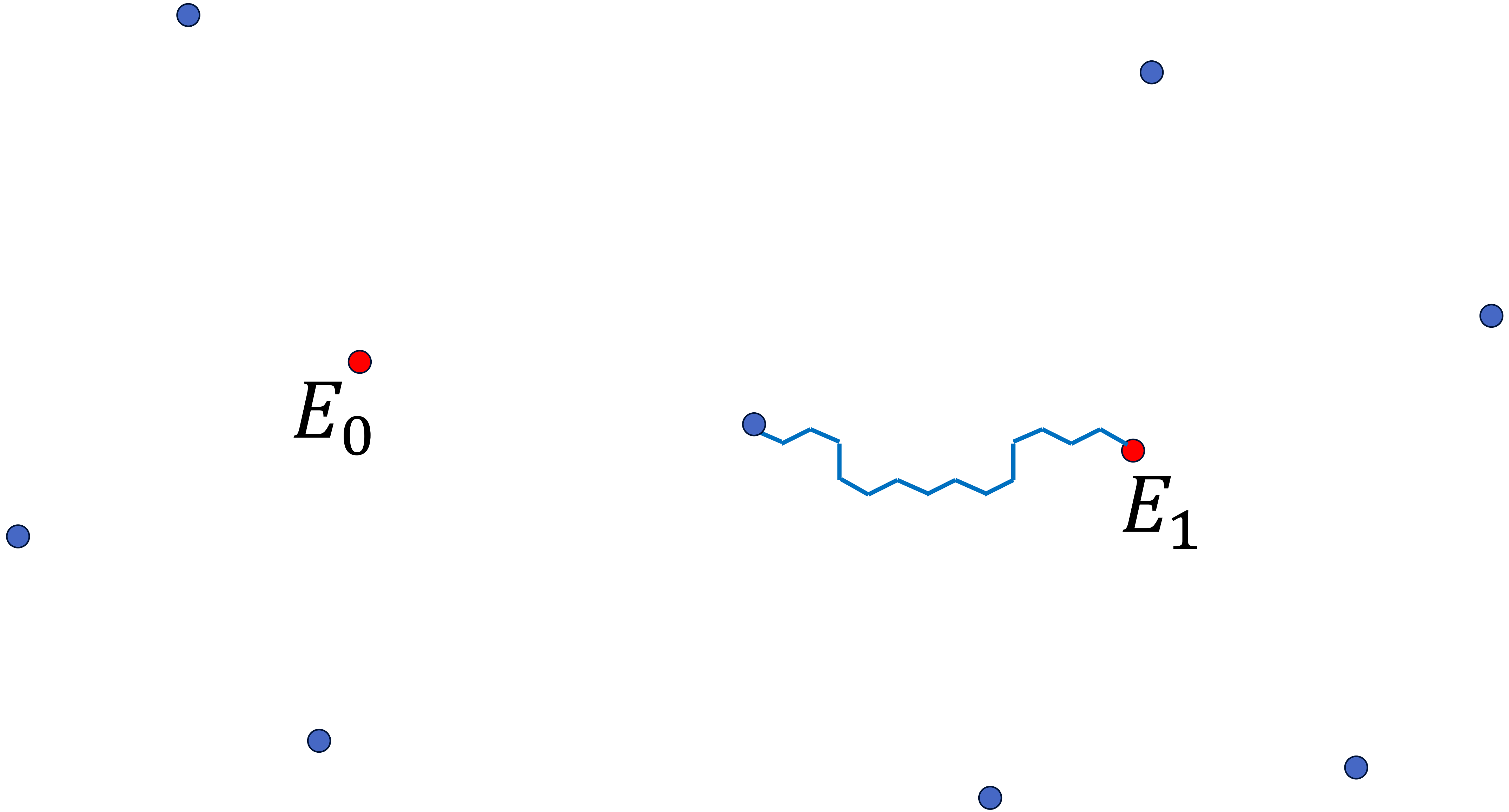

E_0
E_1


E_0
E_1

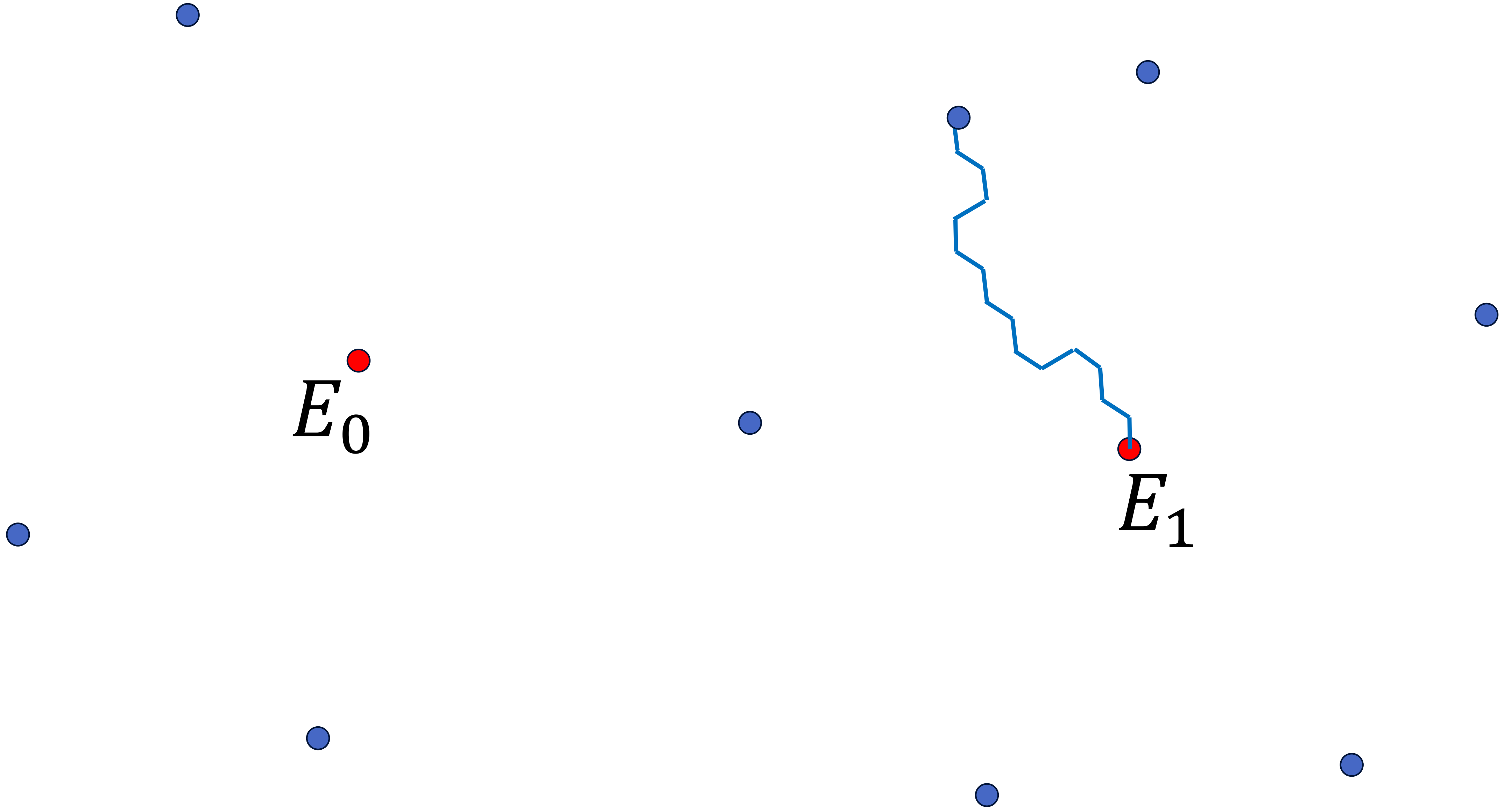

E_0
E_1

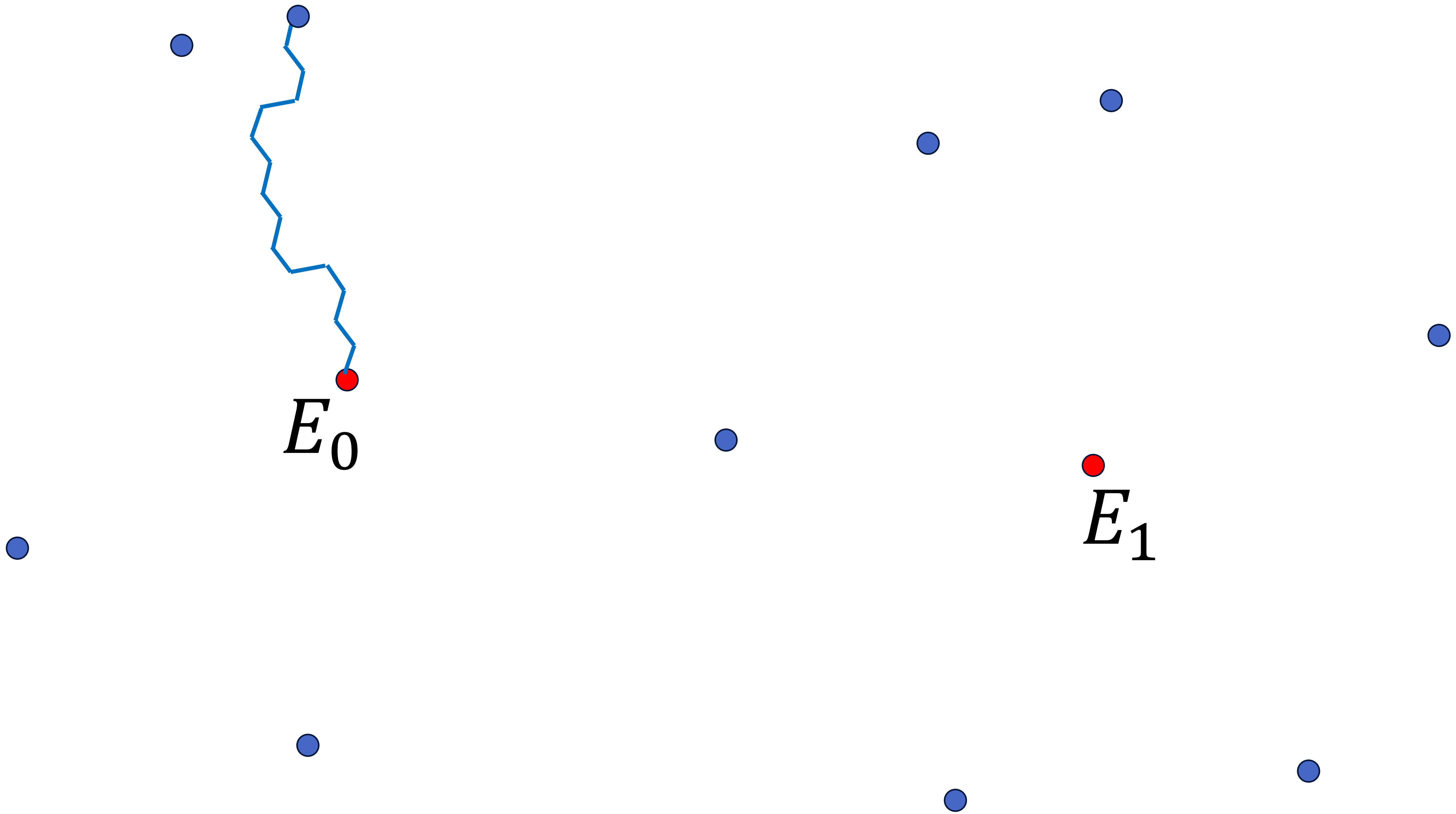

E_0
E_1


E_0
E_1

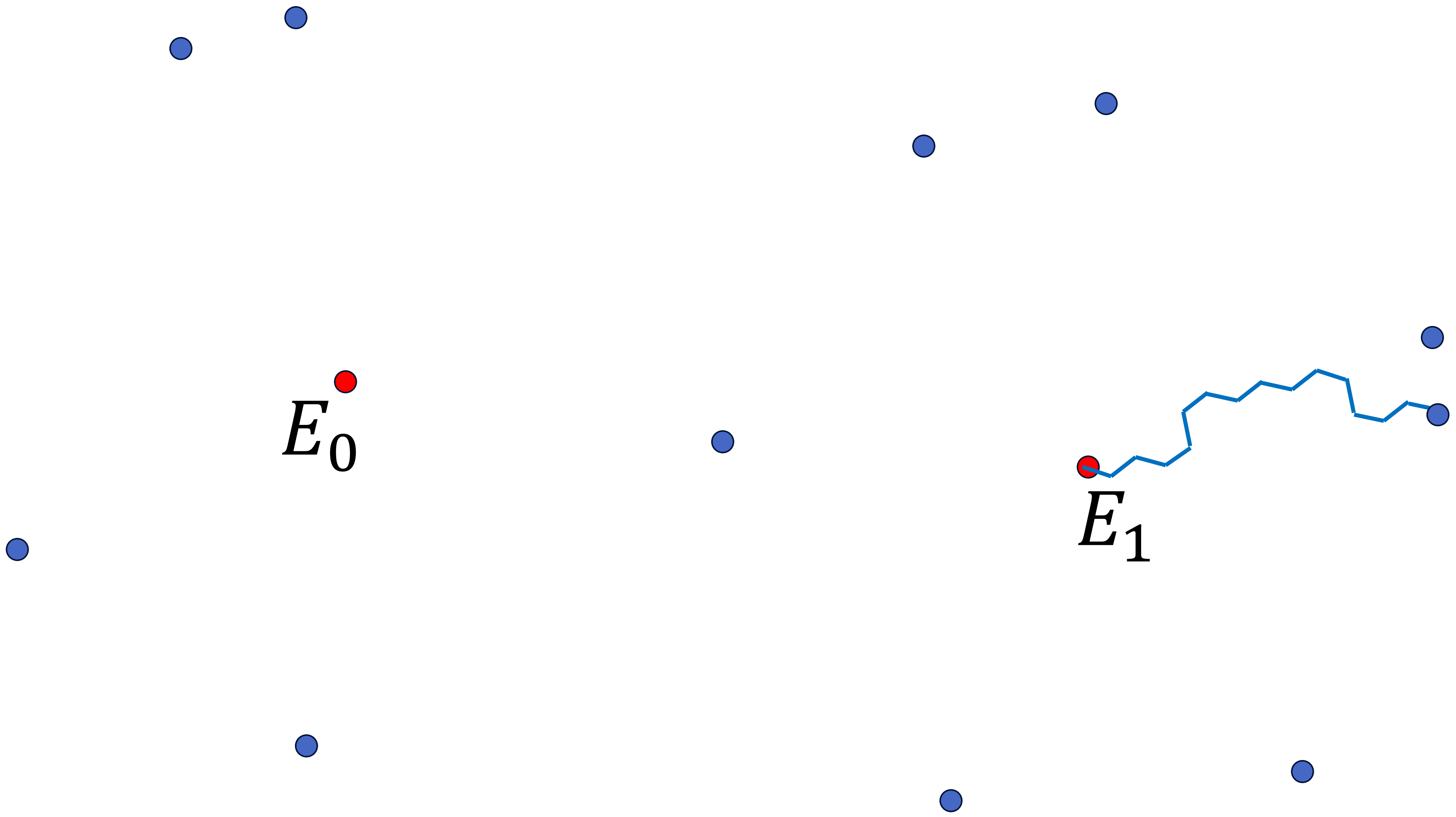


E_0
E_1

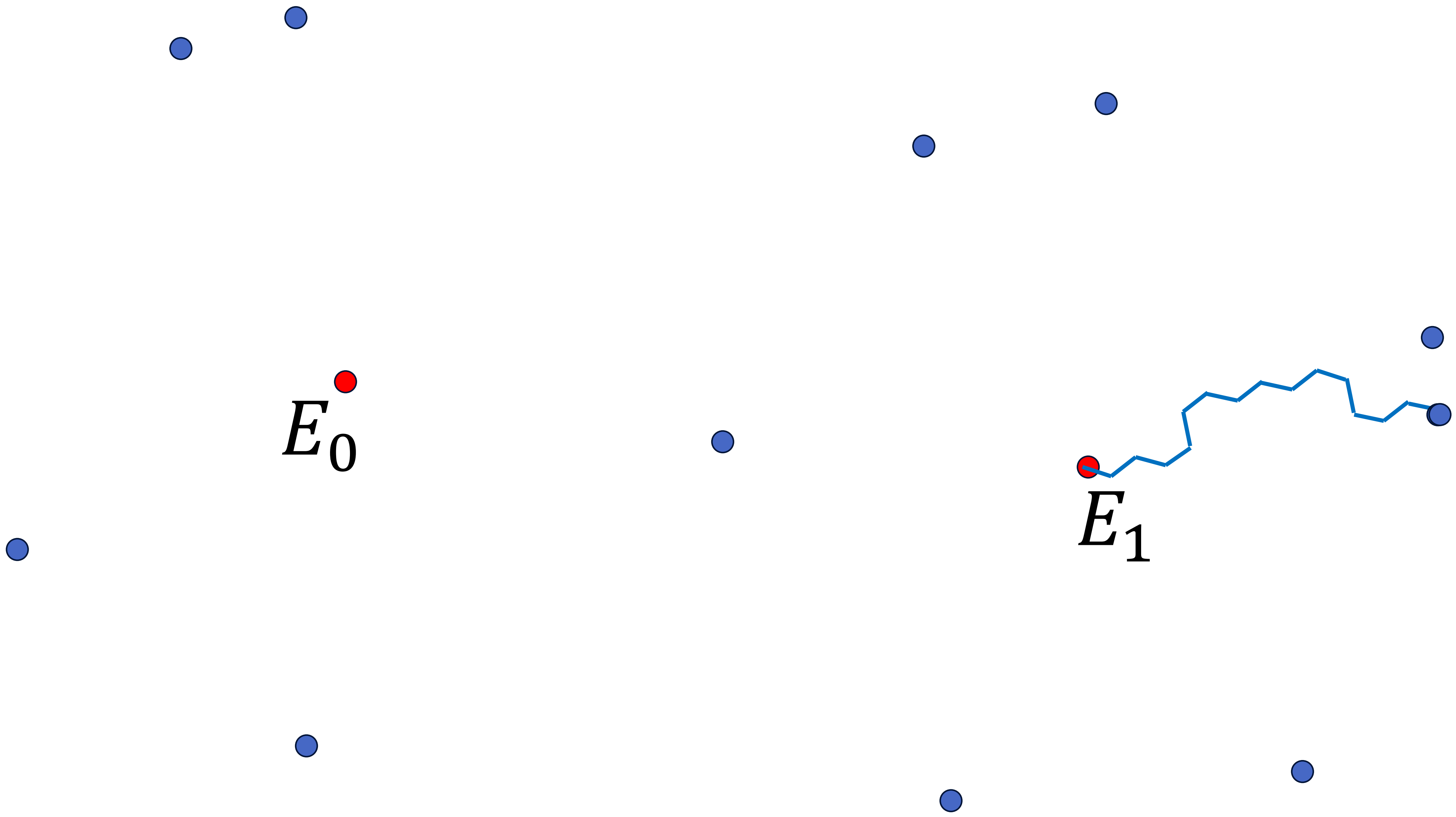

E_0
E_1

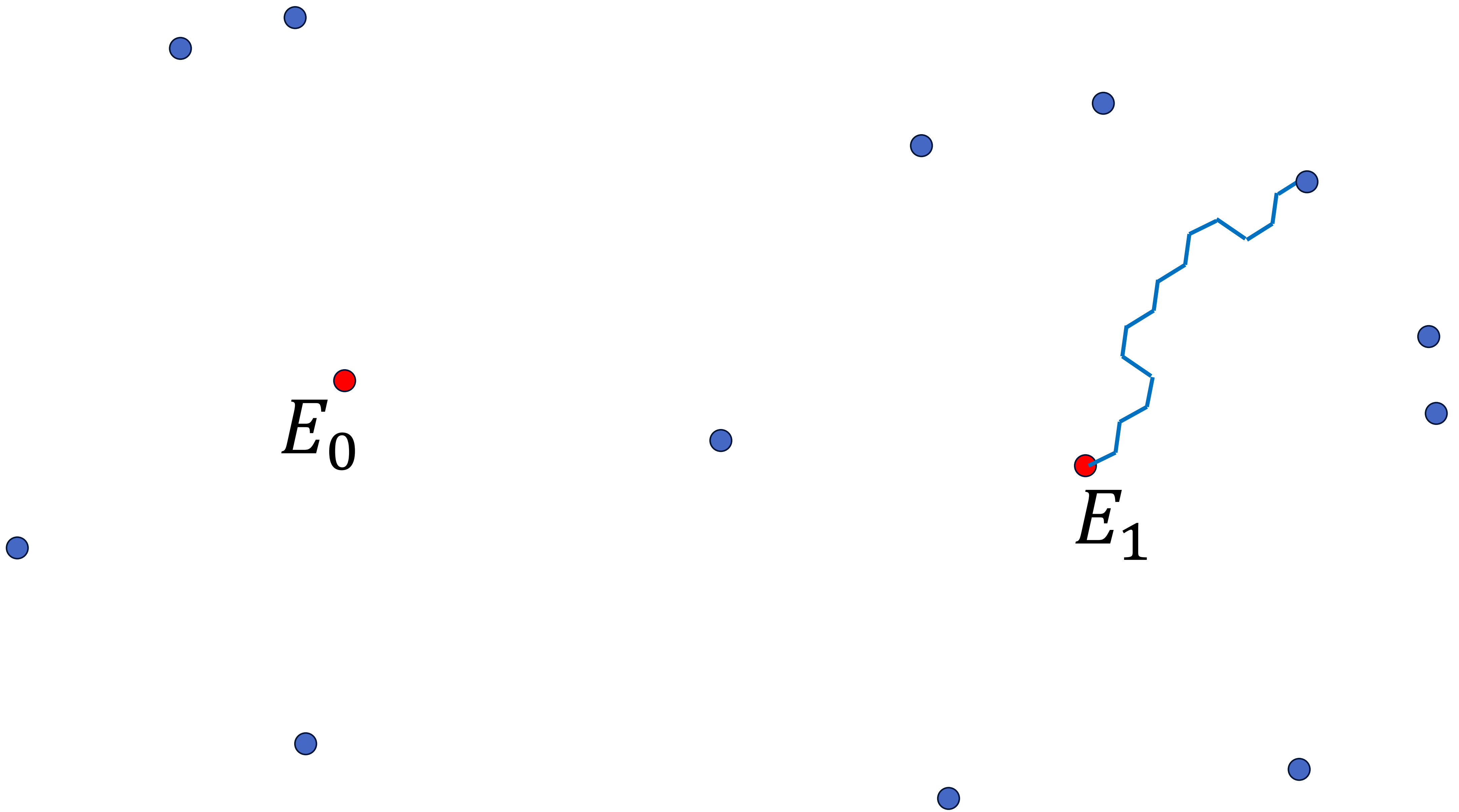


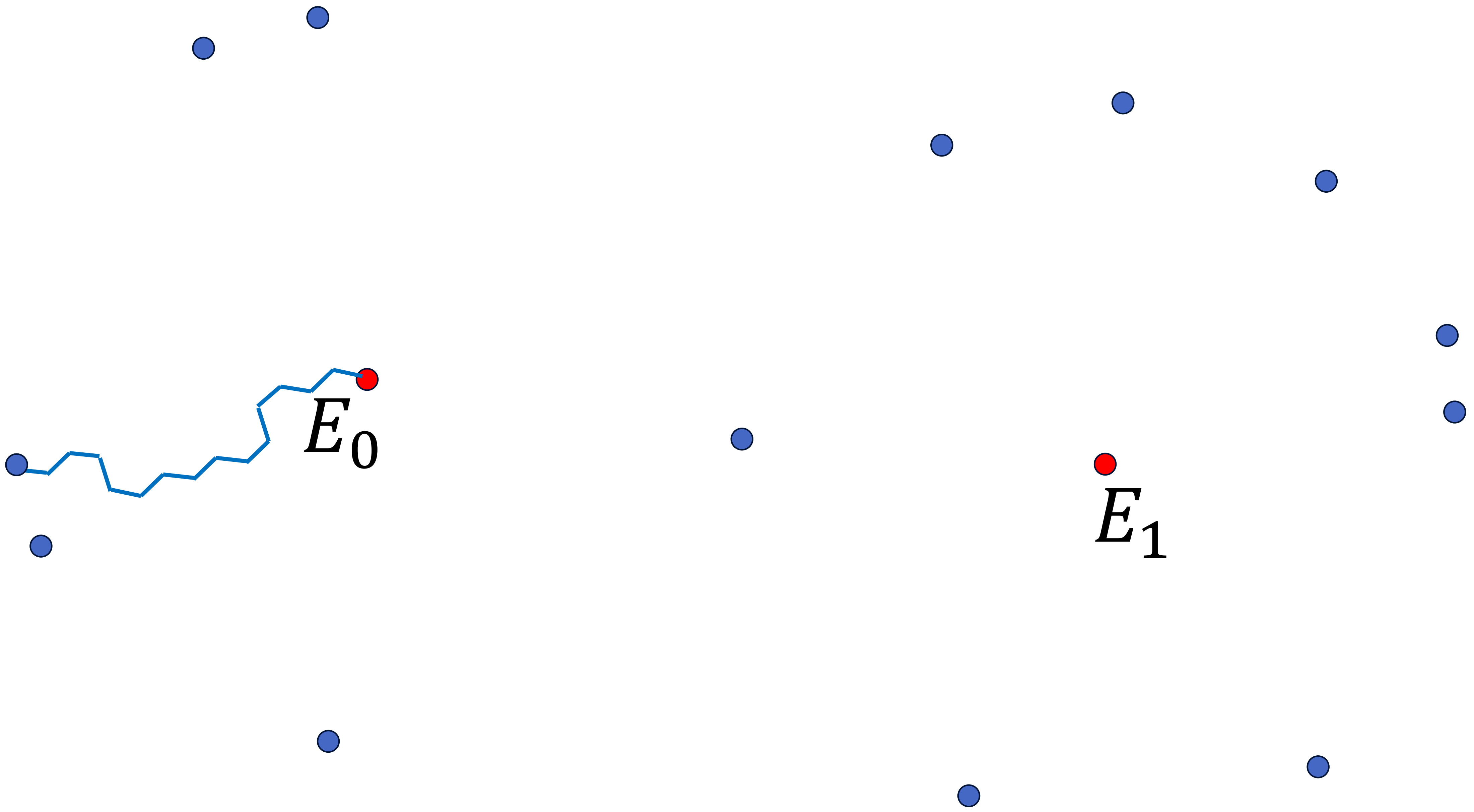
E_0
E_1

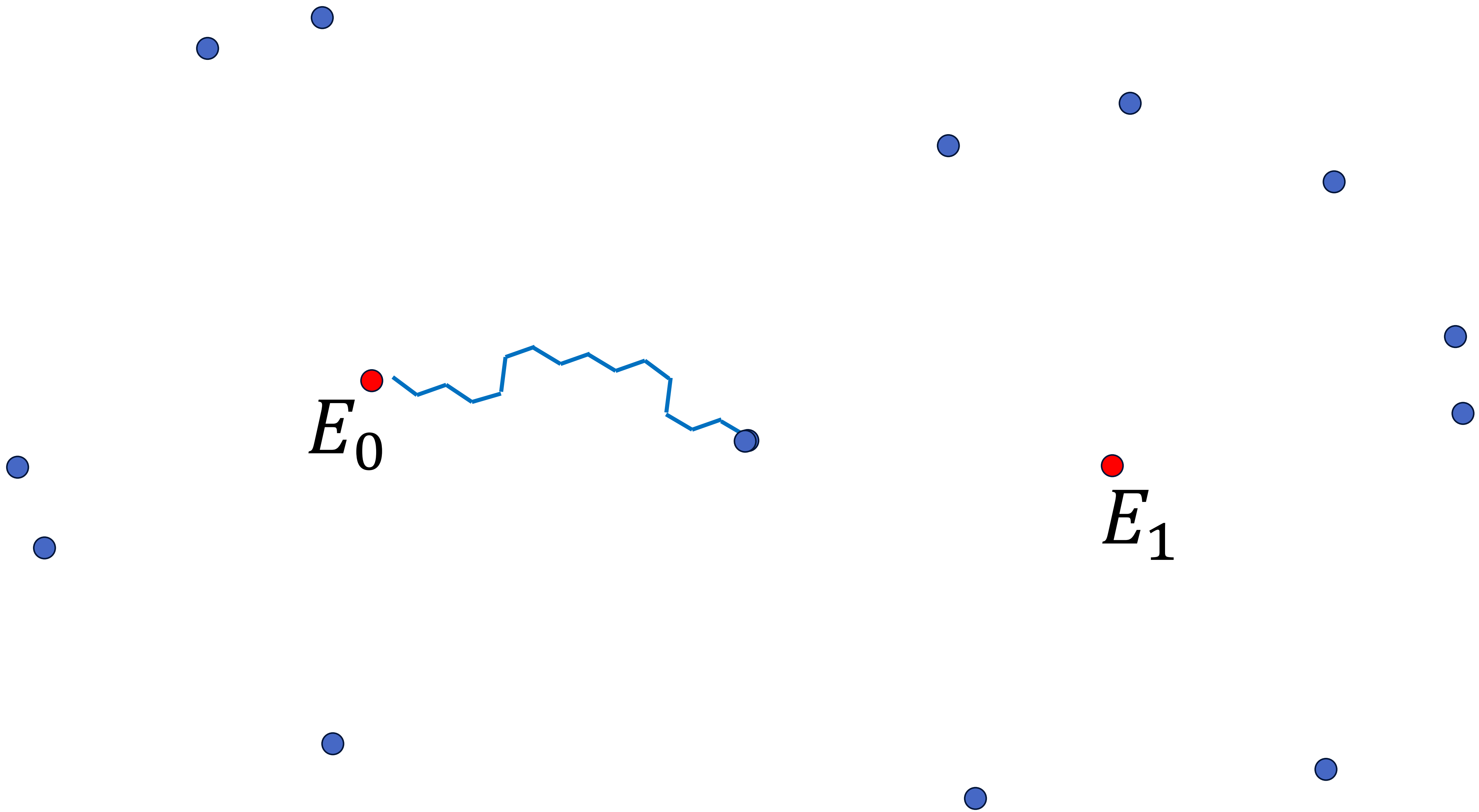


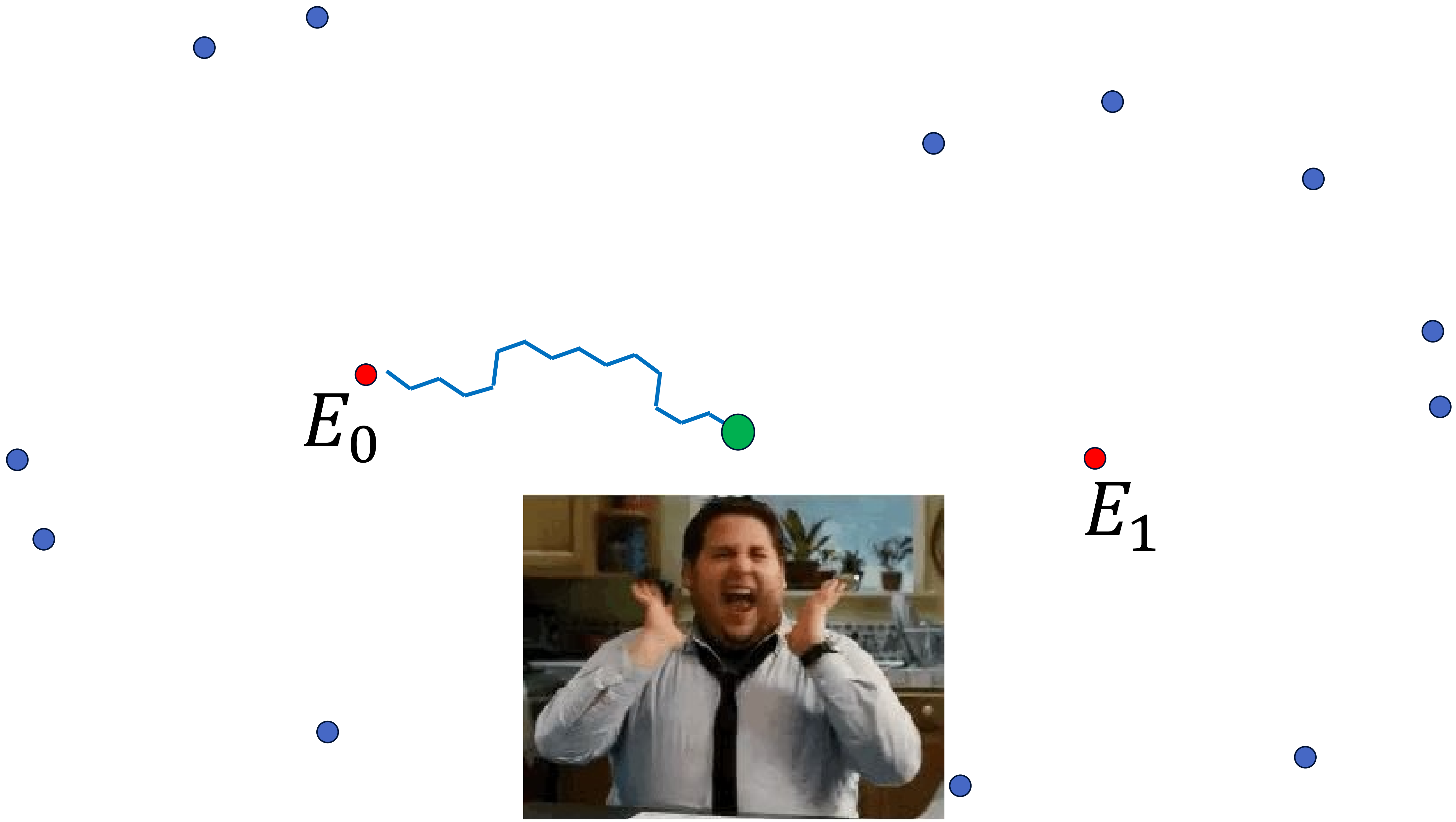
E_0
E_1


E_0
E_1

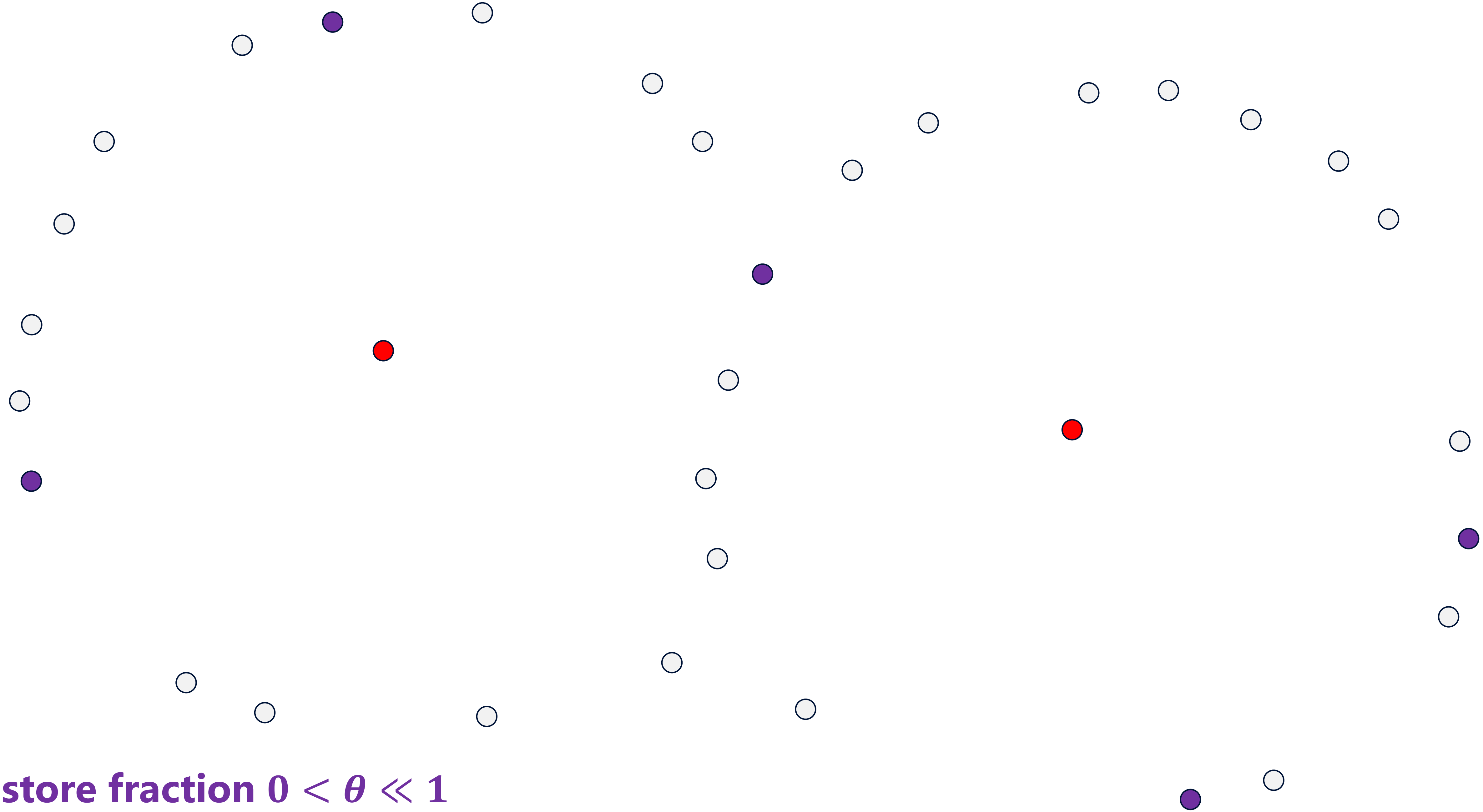

E_0
E_1


E_0
E_1


E_0
E_1


E_0
E_1

can’t possibly store all these: fix w as upper bound on $\#x_i$ storage



store fraction $0 < \theta \ll 1$

vOW

$f_n\!: \quad S \rightarrow S$

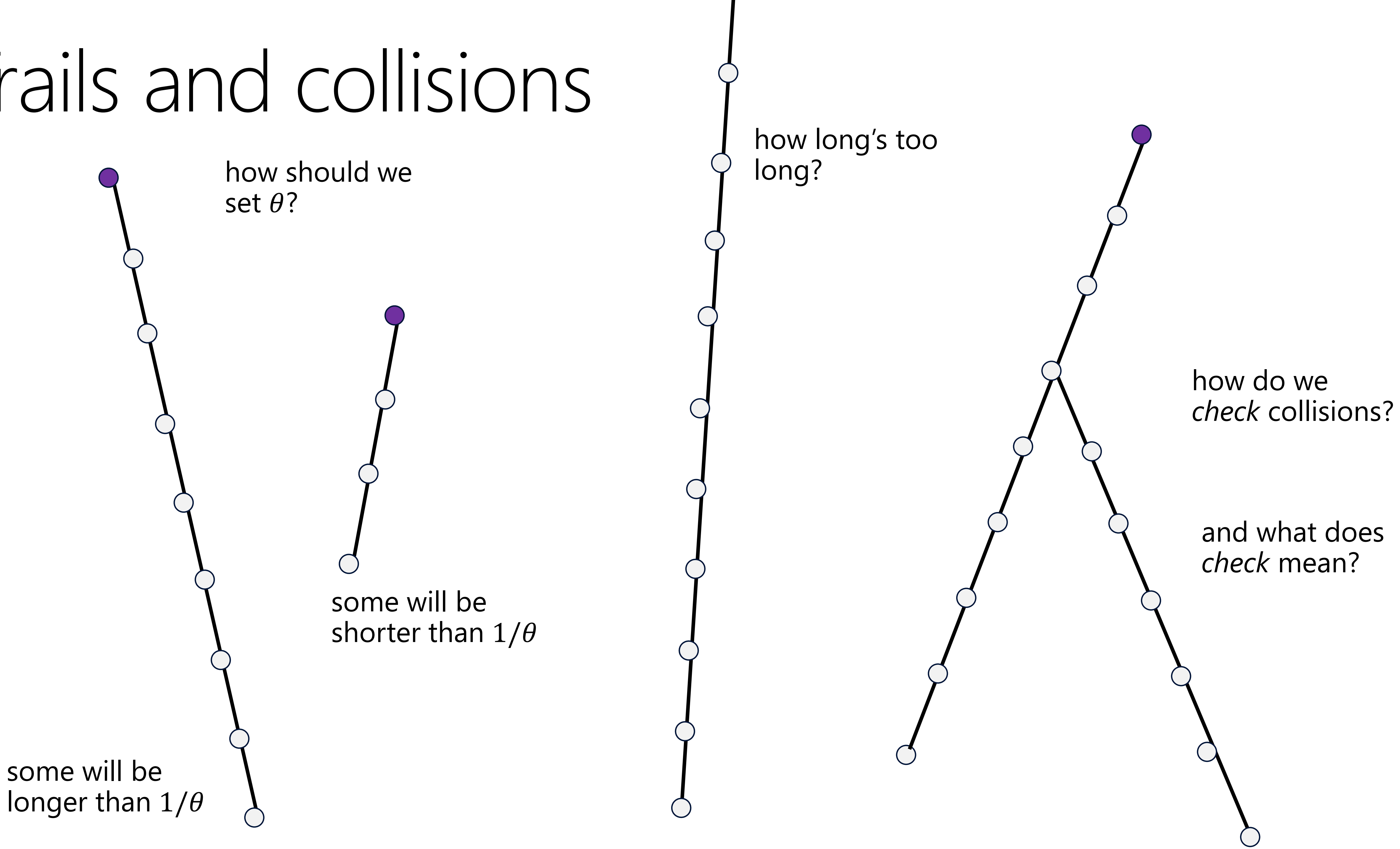
- f_n is a deterministic *random* function, different for each $IV = n$

- For a fixed n, each processor does the following:

  - pick a random starting point x_0
  - produce trail $x_i = f_n(x_{i-1})$, for $i = 1,2 \ldots$
  - stop when x_d is "distinguished" ($1/\theta$).

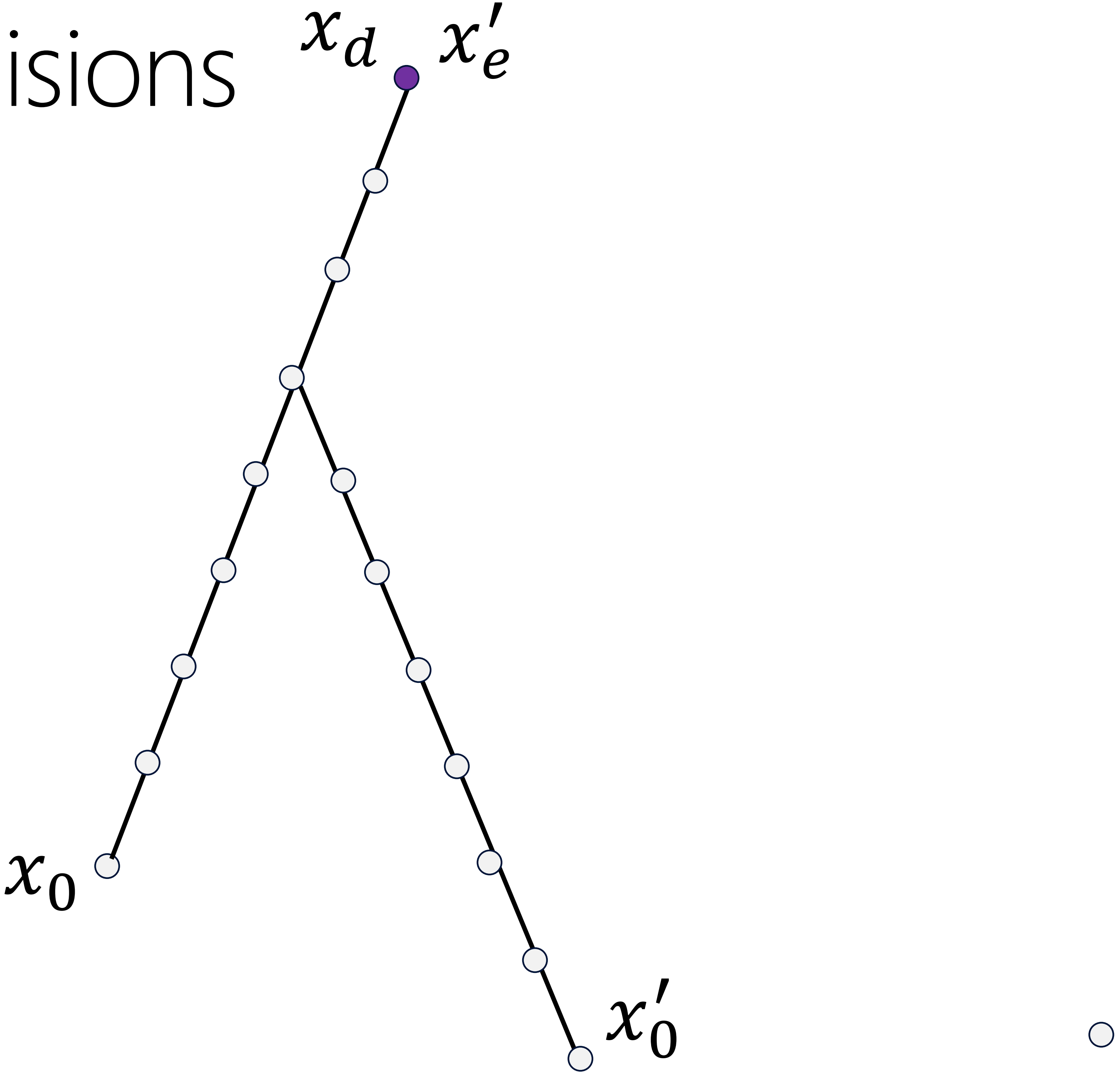
if (x_d has not been seen yet) then
    store triple (x_0, x_d, d) and resample
else
    if (collision not "golden") then
        overwrite previous triple (x_0, x_d, d) and resample
    else 

Trails and collisions



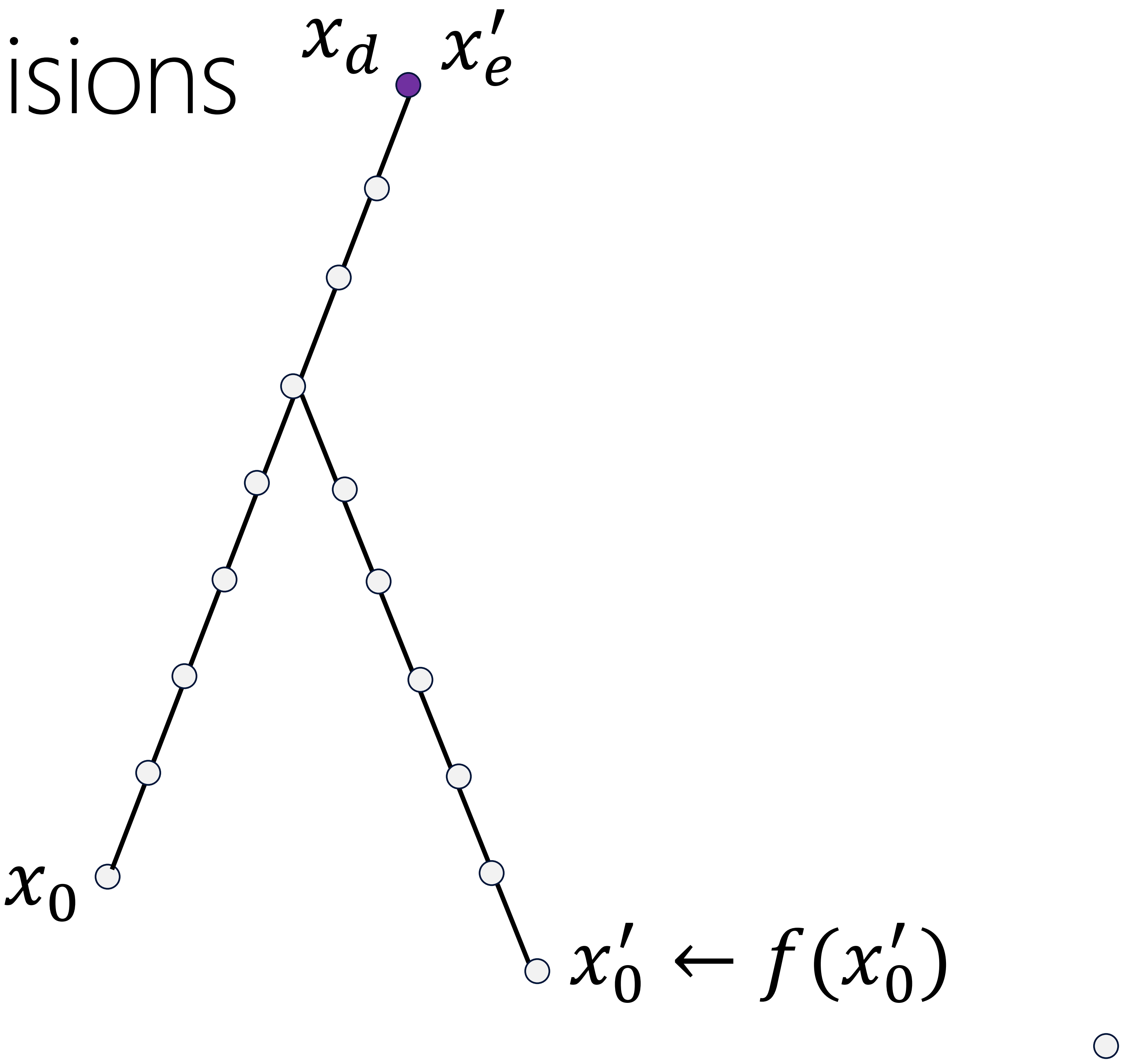
Checking collisions

memory
(x_0, x_d, d)
(x'_0, x'_e, e)



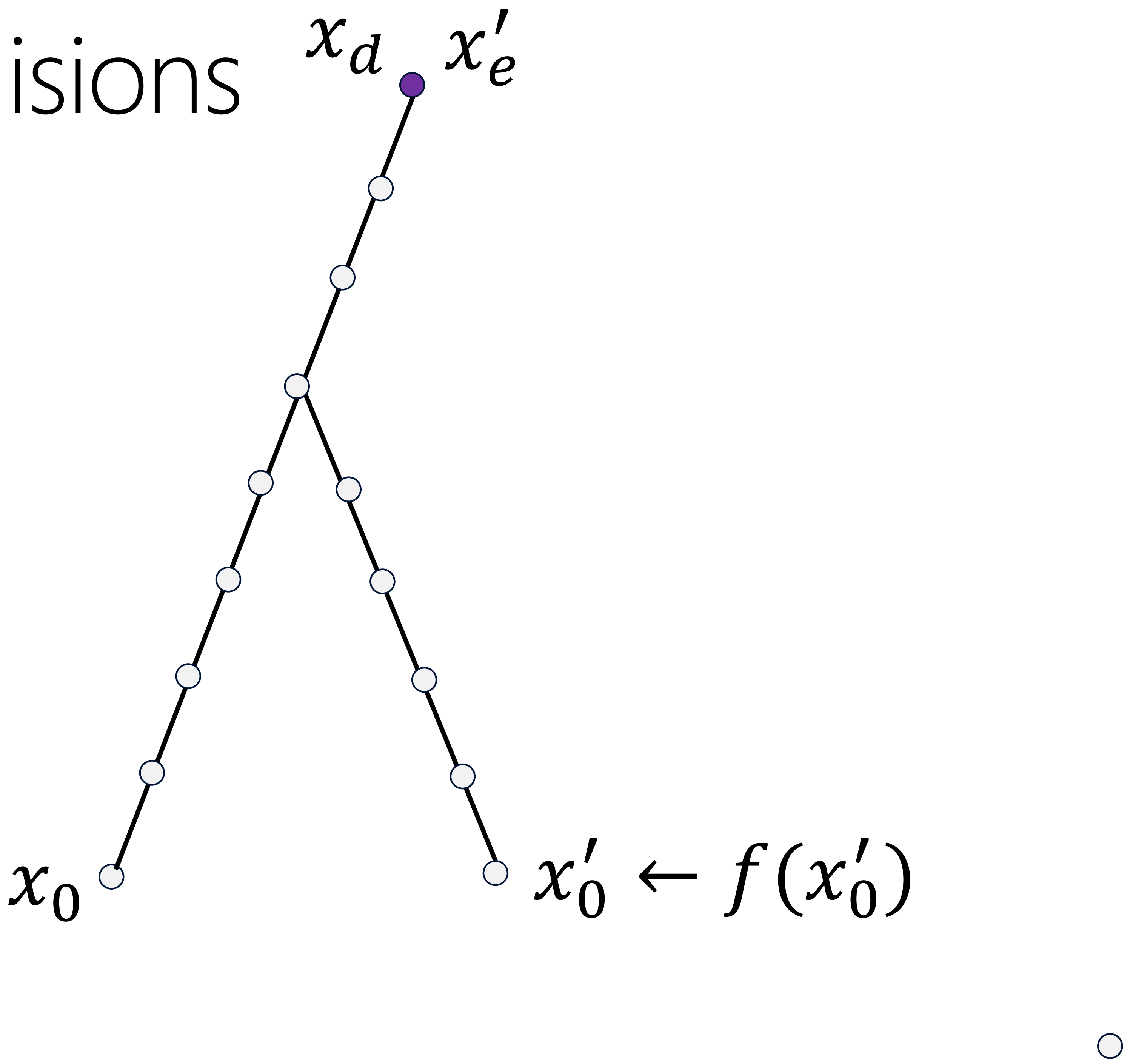
Checking collisions

memory
(x_0, x_d, d)
(x'_0, x'_e, e)



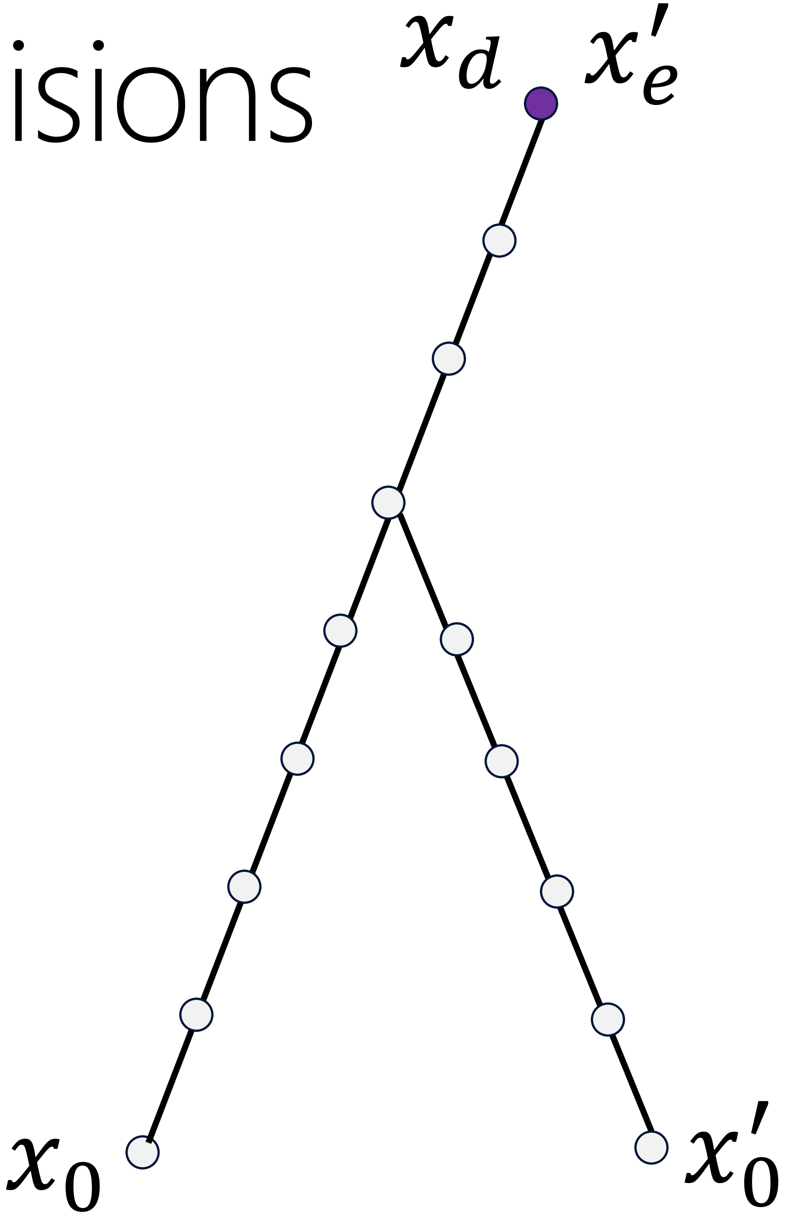
Checking collisions

memory
(x_0, x_d, d)
(x'_0, x'_e, e)



Checking collisions

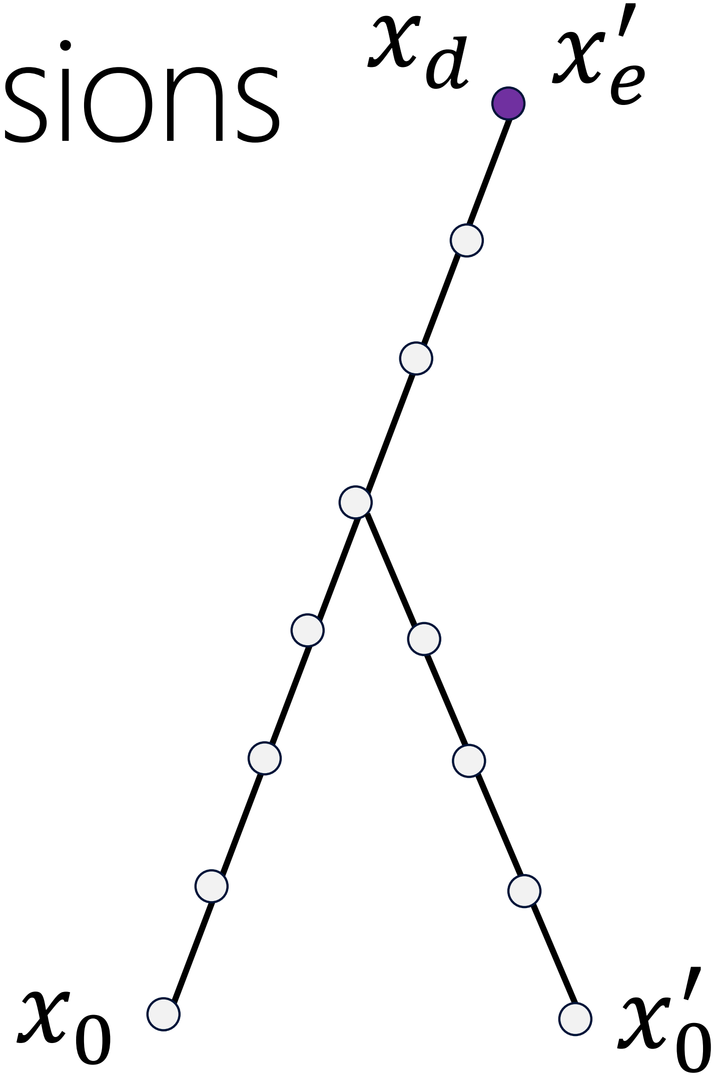
memory
(x_0, x_d, d)
(x'_0, x'_e, e)



$f_n(x_0) \neq f_n(x'_0)$

Checking collisions

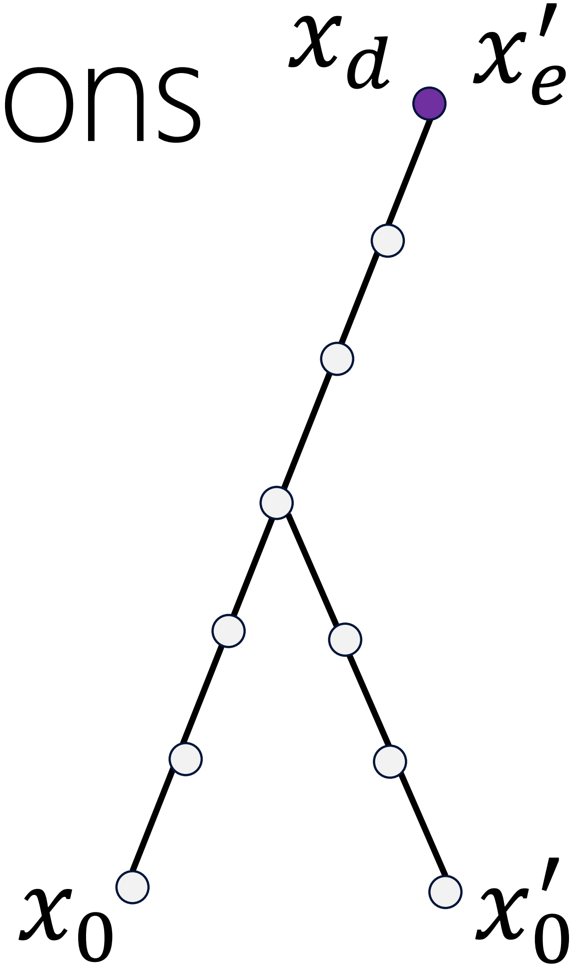
memory
(x_0, x_d, d)
(x'_0, x'_e, e)



$f_n(x_0) \neq f_n(x'_0)$

Checking collisions

memory
(x_0, x_d, d)
(x'_0, x'_e, e)



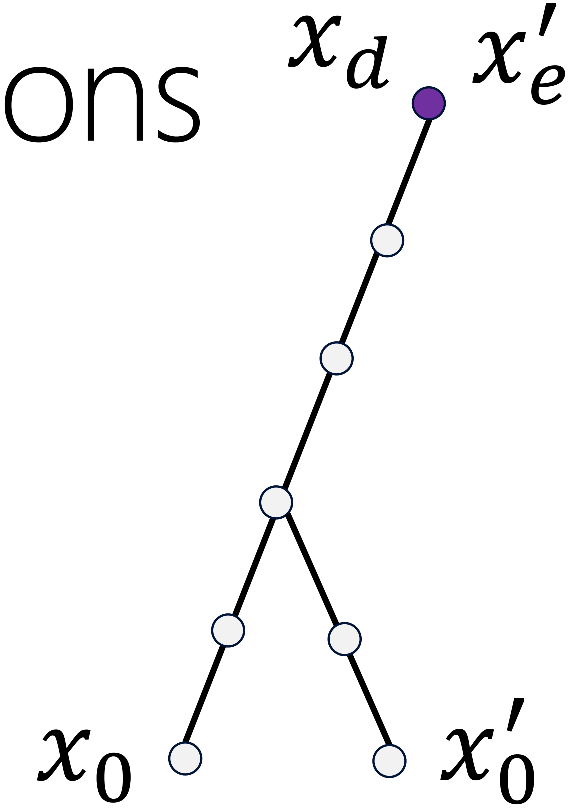
$f_n(x_0) \neq f_n(x'_0)$

Checking collisions

memory

(x_0, x_d, d)

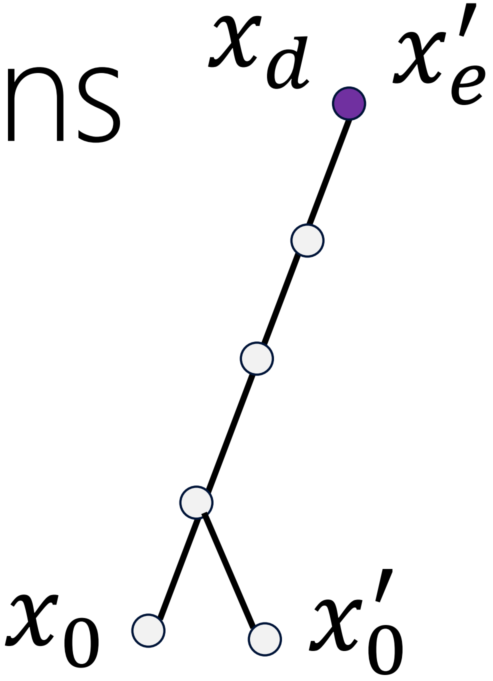
(x'_0, x'_e, e)



$f_n(x_0) \neq f_n(x'_0)$

Checking collisions

memory
(x_0, x_d, d)
(x'_0, x'_e, e)

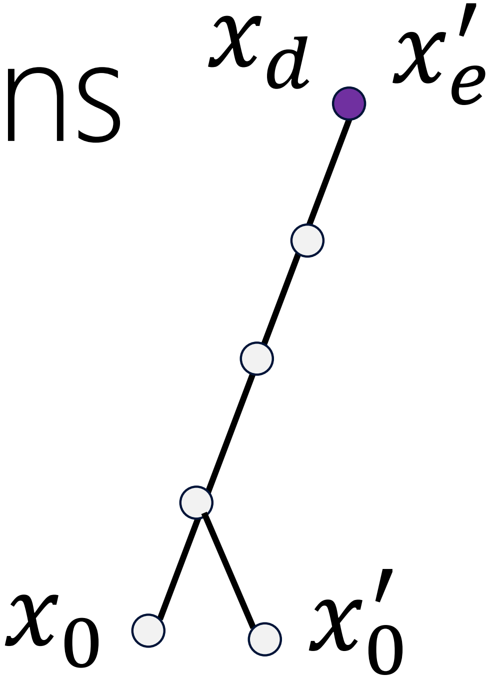


$f_n(x_0) = f_n(x'_0)$
$x_0 \neq x'_0$

DONE?

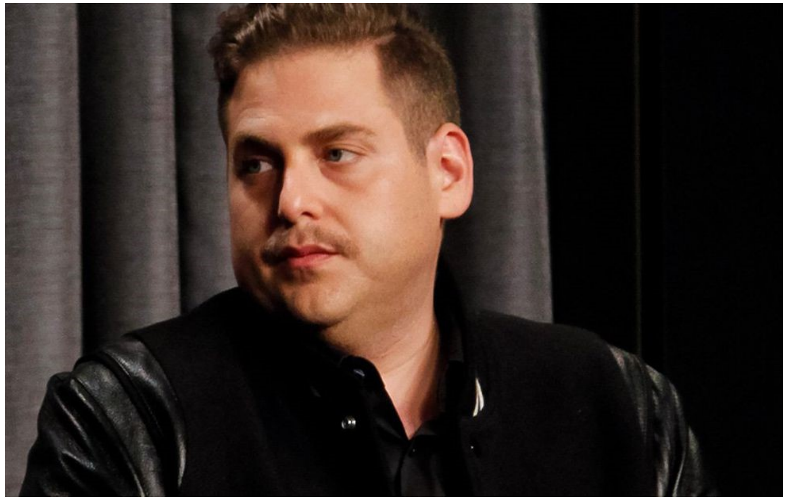
Checking collisions

memory
(x_0, x_d, d)
(x_0', x_e', e)



$f_n(x_0) = f_n(x_0')$

Nope! False alarm



Random collisions vs. the golden collision

- A random function $f_n : \; S \to S$ has many collisions, e.g., think of the random function as a hash function (it kinda is anyway)

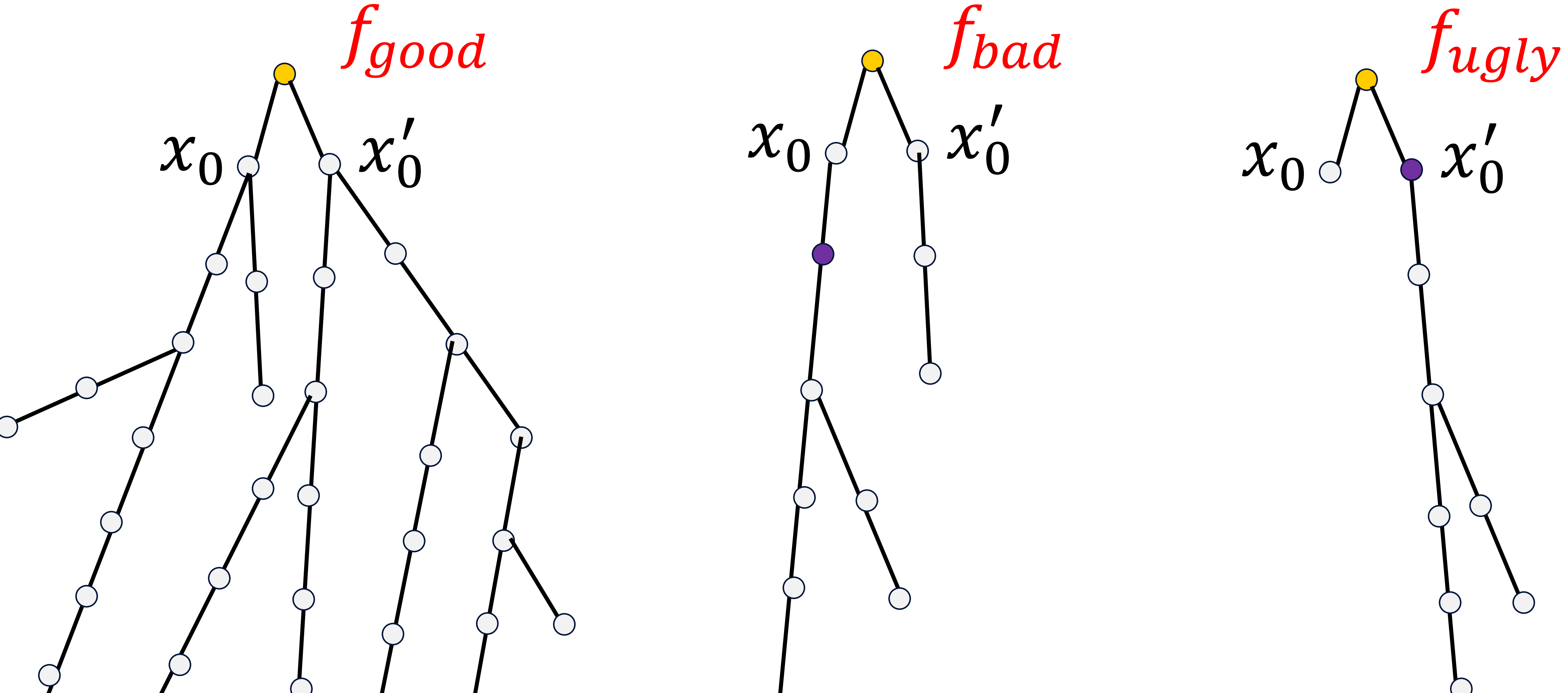
- We will encounter many of these before we hit the one we want, i.e., the "golden collision"

- Much of the algorithm is spent walking, much is spent checking useless annoying collisions

- Ideally there'll be many paths that take us to the golden collision…

Random f_n: the good, the bad and the ugly...

- Even more annoying is that we have to restart the whole algorithm, time and time again...



vOW Complexity

- Analysis conducted by van Oorschot and Wiener
- Analysis confirmed (for CSSI) by Adj *et al.*
- Analysis re-confirmed (for CSSI) by Jaques-Schanck
- Analysis re-re-confirmed (for CSSI) by us

$$T \approx 2.5\sqrt{N^3/w} \cdot t$$

- T = time taken to find golden collision
- $N = |S|$, the number of x_i, approx. $p^{1/4}$
- w = the maximum number of x_i that can be stored.
- t = the time taken to compute $f_n : x_i \mapsto x_{i+1}$ (i.e., half-sized isogeny+ϵ)

vOW security ($w = 2^{80}$)

			2-torsion		3-torsion	
NIST level	Name	(e_A, e_B)	$\log_2(N)$	$\mathbf{log_2(vOW)}$	$\log_2(N)$	$\mathbf{log_2(vOW)}$
1	SIKEp434	$(216,137)$	107	$\mathbf{143}$	107	$\mathbf{144}$
3	SIKEp610	$(305,192)$	151	$\mathbf{210}$	150	$\mathbf{210}$
5	SIKEp751	$(372,239)$	185	$\mathbf{262}$	188	$\mathbf{268}$

$\mathbf{log_2(vOW)}$: count of number of x64 instructions required to mount vOW.
Intended as conservative lower-bound on the classical gate count.

Uncompressed SIKE

	Round 1			Round 2		
NIST level	prime (bits)	PK size (bytes)	cycles (m) (enc+dec)	prime (bits)	PK size (bytes)	Cycles (m) (enc+dec)
1	503	378	30.7	434	326	21.9
3	751	564	88.5	610	458	52.8
5	964	723	-	751	564	88.5

Uncompressed vs. compressed SIKE

	SIKE
Sec. (NIST)	prime (bits)
1	434
3	610
5	751

uncompressed	
PK size (bytes)	Cycles (m) (enc+dec)
326	21.9
458	52.8
564	88.5

compressed	
PK size (bytes)	Cycles (m) (enc+dec)
191	tbd.
268	tbd.
330	tbd.

questions?