Tutorial on Language Generation

Organizers: *Moses Charikar, Anay Mehrotra, Charlotte Peale, Chirag Pabbaraju, Grigoris Velegkas*

Stronger Notions of Generation and Comparison to Prediction

Chirag Pabbaraju

Tutorial on Language Generation in the Limit

COLT 2025

Visit: LanguageGeneration.github.io

Language Generation in the Limit

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

x_1

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

- At each time step t, algorithm generates a string z_t

- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

x_1

z_1

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t

- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

x_1 x_2

z_1

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

	x_1	x_2
	z_1	z_2

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t

- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

x_1	x_2	x_3
z_1	z_2	

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3
😇	z_1	z_2	z_3

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3	x_4
😇	z_1	z_2	z_3	

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t

- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

	x_1	x_2	x_3	x_4
	z_1	z_2	z_3	z_4

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t

- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3	x_4	....
😇	z_1	z_2	z_3	z_4	....

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3	x_4	…………	$x_{t^\star}$
😇	z_1	z_2	z_3	z_4	…………	

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

	x_1	x_2	x_3	x_4	....	$x_{t^\star}$
	z_1	z_2	z_3	z_4	....	$z_{t^\star}$

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

	x_1	x_2	x_3	x_4	....	$x_{t^\star}$
	z_1	z_2	z_3	z_4	....	$z_{t^\star}$ new, $\in L_{70}$

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3	x_4	....	$x_{t^\star}$	$x_{t^\star+1}$
😇	z_1	z_2	z_3	z_4	....	$z_{t^\star}$ new, $\in L_{70}$	

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t

- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3	x_4	....	$x_{t^\star}$	$x_{t^\star+1}$
😇	z_1	z_2	z_3	z_4	....	$z_{t^\star}$ new, $\in L_{70}$	$z_{t^\star+1}$

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3	x_4	....	$x_{t^\star}$	$x_{t^\star+1}$
😇	z_1	z_2	z_3	z_4	....	$z_{t^\star}$ new, $\in L_{70}$	$z_{t^\star+1}$ new, $\in L_{70}$

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

	x_1	x_2	x_3	x_4	…………	$x_{t^\star}$	$x_{t^\star+1}$	…………
	z_1	z_2	z_3	z_4	…………	$z_{t^\star}$ new, $\in L_{70}$	$z_{t^\star+1}$ new, $\in L_{70}$	…………

Language Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t

$t^\star$ can depend on target L_z as well as enumeration order! 😓

- Generates in the limit if beyond some large enough $t^\star$, all strings generated are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3	x_4	.....	$x_{t^\star}$	$x_{t^\star+1}$	.....
😇	z_1	z_2	z_3	z_4	.....	$z_{t^\star}$ new, $\in L_{70}$	$z_{t^\star+1}$ new, $\in L_{70}$	.....

Language Generation in the Limit

Language Generation in the Limit

- Limitation: Definition allows $t^\star$ beyond which algorithm generates validly to depend on the enumeration order!

Language Generation in the Limit

- Limitation: Definition allows $t^\star$ beyond which algorithm generates validly to depend on the enumeration order!

- Example: Suppose $\mathcal{C} = \{L_1, L_2\}$

$$L_1 = \{\dots, -3, -2, -1, 1, 2, 3, \dots\}$$

$$L_2 = \{0, 1, 2, 3, 4, \dots\}$$

Language Generation in the Limit

- Limitation: Definition allows $t^\star$ beyond which algorithm generates validly to depend on the enumeration order!

- Example: Suppose $\mathcal{C} = \{L_1, L_2\}$

$$L_1 = \{\dots, -3, -2, -1, 1, 2, 3, \dots\}$$
$$L_2 = \{0, 1, 2, 3, 4, \dots\}$$

- Suppose L_2 is the target language, but adversary enumerates it as

$$1, 2, 3, 4, 5, 6, \dots$$

Language Generation in the Limit

- Limitation: Definition allows $t^\star$ beyond which algorithm generates validly to depend on the enumeration order!

- Example: Suppose $\mathcal{C} = \{L_1, L_2\}$

$$L_1 = \{\dots, -3, -2, -1, 1, 2, 3, \dots\}$$

$$L_2 = \{0, 1, 2, 3, 4, \dots\}$$

- Suppose L_2 is the target language, but adversary enumerates it as

$$1, 2, 3, 4, 5, 6, \dots$$

- Natural algorithm: generate from first consistent language in $\mathcal{C}$

Language Generation in the Limit

- Limitation: Definition allows $t^\star$ beyond which algorithm generates validly to depend on the enumeration order!

- Example: Suppose $\mathcal{C} = \{L_1, L_2\}$

$$L_1 = \{\dots, -3, -2, -1, 1, 2, 3, \dots\}$$
$$L_2 = \{0, 1, 2, 3, 4, \dots\}$$

- Suppose L_2 is the target language, but adversary enumerates it as

$$1, 2, 3, 4, 5, 6, \dots$$

- Natural algorithm: generate from first consistent language in $\mathcal{C}$

- Until adversary shows 0, can keep generating negative numbers from L_1 😓

Language Generation in the Limit

- Limitation: Definition allows $t^\star$ beyond which algorithm generates validly to depend on the enumeration order!

- Example: Suppose $\mathcal{C} = \{L_1, L_2\}$

$$L_1 = \{\dots, -3, -2, -1, 1, 2, 3, \dots\}$$
$$L_2 = \{0, 1, 2, 3, 4, \dots\}$$

Kleinberg-Mullainathan's algorithm also faces this issue 😓

- Suppose L_2 is the target language, but adversary enumerates it as

$$1, 2, 3, 4, 5, 6, \dots$$

- Natural algorithm: generate from first consistent language in $\mathcal{C}$

- Until adversary shows 0, can keep generating negative numbers from L_1 😓

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

😈	x_1	x_2	x_3	x_4	....	$x_{t^\star}$	$x_{t^\star+1}$	....
😇	z_1	z_2	z_3	z_4	....	$z_{t^\star}$ new, $\in L_{70}$	$z_{t^\star+1}$ new, $\in L_{70}$	....

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots$$

- At each time step t, algorithm generates a string z_t

- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

x_1' x_2' x_3' x_4' ....

z_1' z_2' z_3' z_4' ....

Non-uniform Generation in the Limit

Li, Raman, Tewari ‘24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

😈 x'_1 x'_2 x'_3 x'_4 ………… $x'_{t^\star}$

😇 z'_1 z'_2 z'_3 z'_4 …………

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

x'_1 x'_2 x'_3 x'_4 .... $x'_{t^\star}$

z'_1 z'_2 z'_3 z'_4 .... $z'_{t^\star}$

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

x'_1 $\quad$ x'_2 $\quad$ x'_3 $\quad$ x'_4 $\quad$ .... $\quad$ $x'_{t^\star}$

z'_1 $\quad$ z'_2 $\quad$ z'_3 $\quad$ z'_4 $\quad$ .... $\quad$ $z'_{t^\star}$

new, $\in L_{70}$

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

😈 x_1' x_2' x_3' x_4' .... $x_{t^\star}'$ $x_{t^\star+1}'$

😇 z_1' z_2' z_3' z_4' .... $z_{t^\star}'$

new, $\in L_{70}$

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

	x_1'	x_2'	x_3'	x_4'	....	$x_{t^\star}'$	$x_{t^\star+1}'$
	z_1'	z_2'	z_3'	z_4'	....	$z_{t^\star}'$ new, $\in L_{70}$	$z_{t^\star+1}'$

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

😈	x'_1	x'_2	x'_3	x'_4	…………	$x'_{t^\star}$	$x'_{t^\star+1}$
😇	z'_1	z'_2	z'_3	z'_4	…………	$z'_{t^\star}$ new, $\in L_{70}$	$z'_{t^\star+1}$ new, $\in L_{70}$

Non-uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

x'_1 x'_2 x'_3 x'_4 .... $x'_{t^\star}$ $x'_{t^\star+1}$ ....

z'_1 z'_2 z'_3 z'_4 .... $z'_{t^\star}$ $z'_{t^\star+1}$ ....

new, $\in L_{70}$ new, $\in L_{70}$

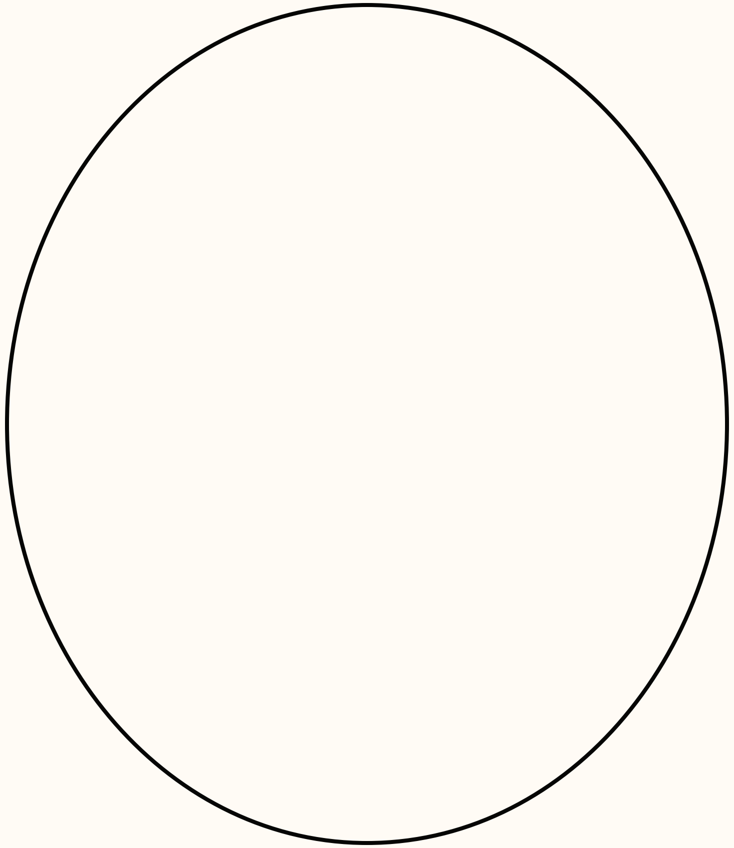
Non-uniform Generation in the Limit

Non-uniform Generation in the Limit

- Question (Li, Raman, Tewari '24): Is every countable collection of languages non-uniformly generatable in the limit?

Non-uniform Generation in the Limit

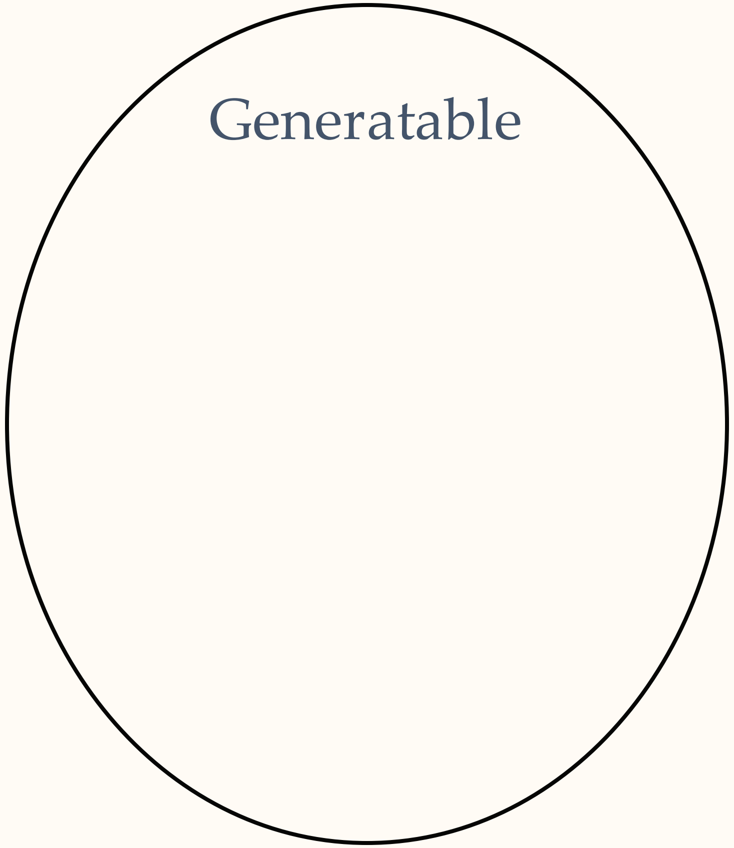
- Question (Li, Raman, Tewari '24): Is every countable collection of languages non-uniformly generatable in the limit?



Countable language collections

Non-uniform Generation in the Limit

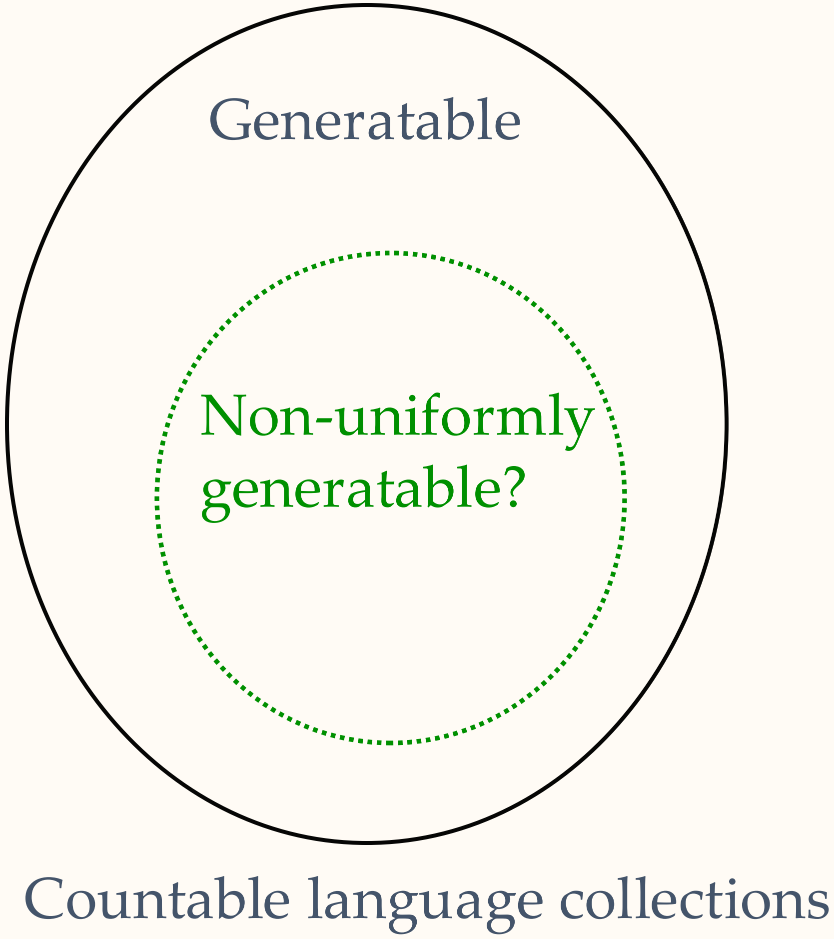
- <u>Question (Li, Raman, Tewari '24):</u> Is every countable collection of languages non-uniformly generatable in the limit?



Countable language collections

Non-uniform Generation in the Limit

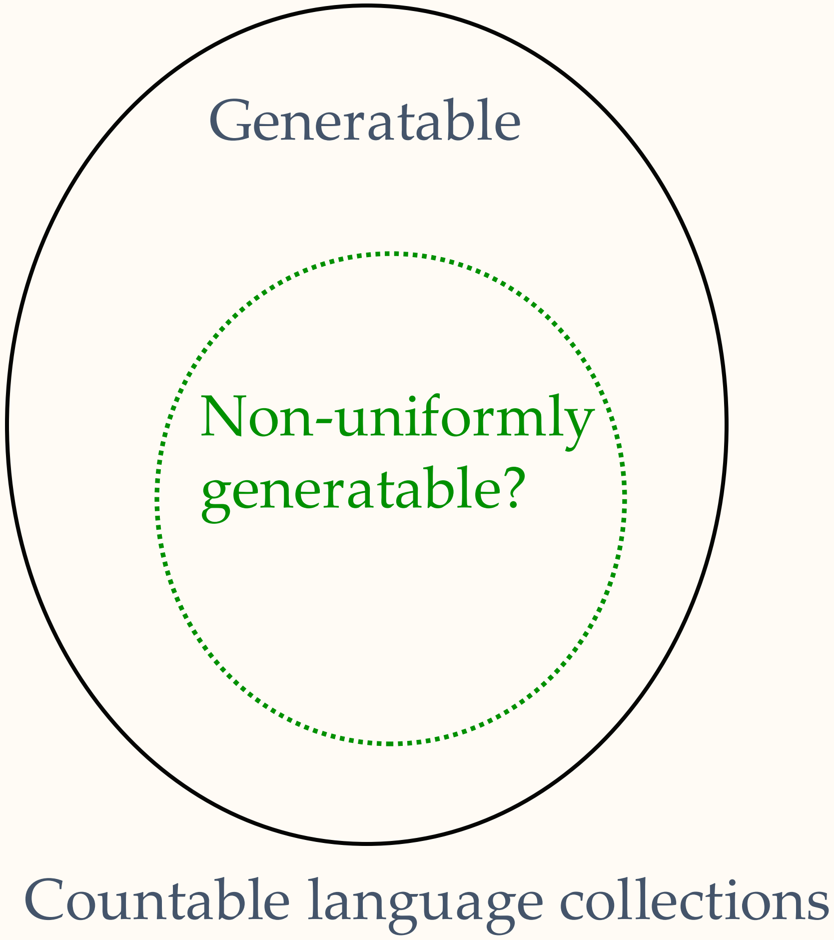
- Question (Li, Raman, Tewari '24): Is every countable collection of languages non-uniformly generatable in the limit?



Non-uniform Generation in the Limit

- Question (Li, Raman, Tewari '24): Is every countable collection of languages non-uniformly generatable in the limit?

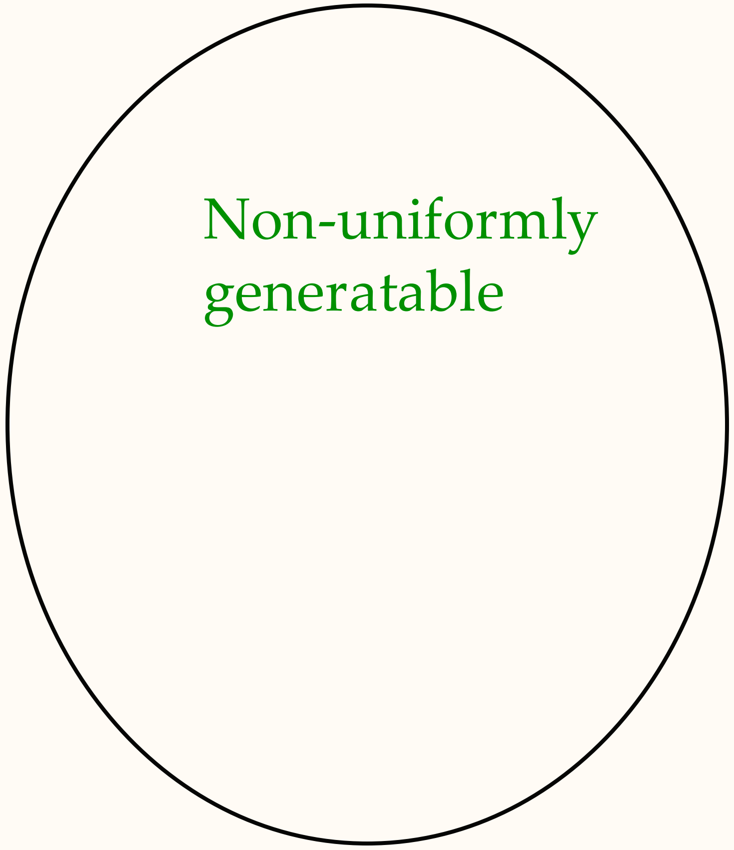
- Theorem (Charikar, P '24, Li, Raman, Tewari '24): Yes! 🙃



Non-uniform Generation in the Limit

- Question (Li, Raman, Tewari '24): Is every countable collection of languages non-uniformly generatable in the limit?

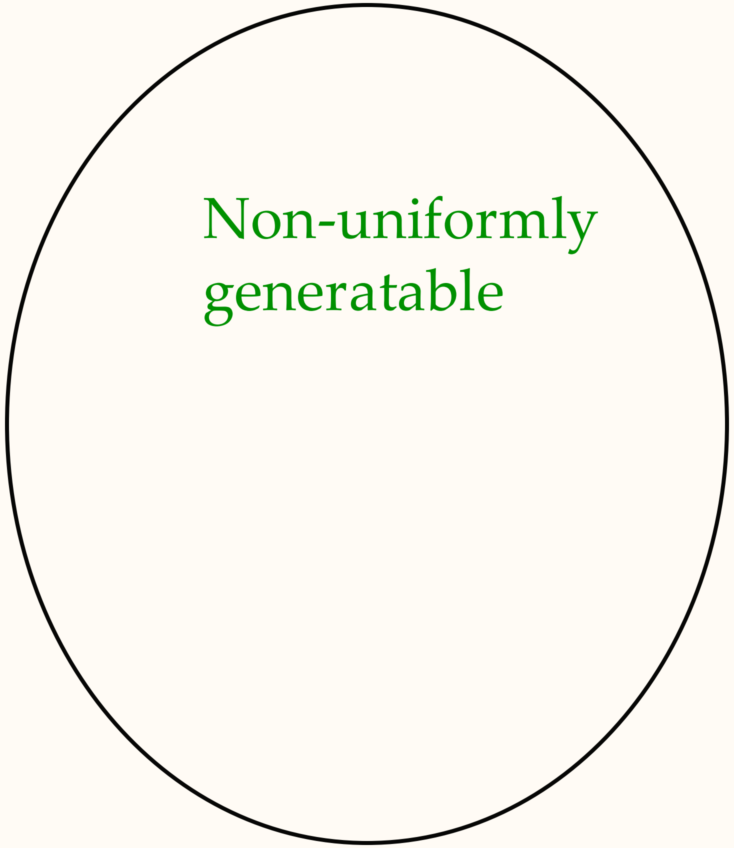
- Theorem (Charikar, P '24, Li, Raman, Tewari '24): Yes! 🙃



Countable language collections

Non-uniform Generation in the Limit

- Question (Li, Raman, Tewari '24): Is every countable collection of languages non-uniformly generatable in the limit?
- Theorem (Charikar, P '24, Li, Raman, Tewari '24): Yes! 🙃
- However, provably requires stronger oracles



Countable language collections

Non-uniform Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

😈	x'_1	x'_2	x'_3	x'_4	....	$x'_{t^\star}$	$x'_{t^\star+1}$	....
😇	z'_1	z'_2	z'_3	z'_4	....	$z'_{t^\star}$ new, $\in L_{70}$	$z'_{t^\star+1}$ new, $\in L_{70}$	....

Non-uniform Generation in the Limit

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

Can we further get $t^\star$ to depend only on $\mathcal{C}$? 🤔

- At each time step t, algorithm generates a string z_t
- Non-uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C}, L_z)$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

😈 x'_1 x'_2 x'_3 x'_4 .... $x'_{t^\star}$ $x'_{t^\star+1}$ ....

😇 z'_1 z'_2 z'_3 z'_4 .... $z'_{t^\star}$ $z'_{t^\star+1}$ ....

new, $\in L_{70}$ new, $\in L_{70}$

Uniform Generation in the Limit

Li, Raman, Tewari ‘24

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

- At each time step t, algorithm generates a string z_t

Uniform Generation in the Limit

Li, Raman, Tewari ‘24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots\dots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{70}

	x_1	x_2	x_3	x_4	......	$x_{t^\star}$	$x_{t^\star+1}$ ......
	z_1	z_2	z_3	z_4	......	$z_{t^\star}$ new, $\in L_{70}$	$z_{t^\star+1}$ ...... new, $\in L_{70}$

Uniform Generation in the Limit

Li, Raman, Tewari ‘24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

😈

😇

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{80}

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{80}

x_1' x_2' x_3' x_4' ....

z_1' z_2' z_3' z_4' ....

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \dots\dots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{80}

😈 x'_1 x'_2 x'_3 x'_4 .... $x'_{t^\star}$

😇 z'_1 z'_2 z'_3 z'_4 ....

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{80}

😈	x_1'	x_2'	x_3'	x_4'	…………	$x_{t^\star}'$
😇	z_1'	z_2'	z_3'	z_4'	…………	$z_{t^\star}'$

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{80}

😈 x'_1 x'_2 x'_3 x'_4 ………… $x'_{t^\star}$

😇 z'_1 z'_2 z'_3 z'_4 ………… $z'_{t^\star}$

new, $\in L_{80}$

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{80}

😈 x'_1 x'_2 x'_3 x'_4 .... $x'_{t^\star}$ $x'_{t^\star+1}$

😇 z'_1 z'_2 z'_3 z'_4 .... $z'_{t^\star}$

new, $\in L_{80}$

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots.$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{80}

x'_1 x'_2 x'_3 x'_4 .... $x'_{t^\star}$ $x'_{t^\star+1}$

z'_1 z'_2 z'_3 z'_4 .... $z'_{t^\star}$ $z'_{t^\star+1}$

new, $\in L_{80}$

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings , all strings generated thereafter are new and in L_z

chooses L_{80}

x'_1	x'_2	x'_3	x'_4	....	$x'_{t^\star}$	$x'_{t^\star+1}$
z'_1	z'_2	z'_3	z'_4	....	$z'_{t^\star}$ new, $\in L_{80}$	$z'_{t^\star+1}$ new, $\in L_{80}$

Uniform Generation in the Limit

Li, Raman, Tewari '24

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

- Adversary chooses some target language L_z, starts enumerating it in an order of their choosing

$$x_1, x_2, x_3, x_4, x_5, \ldots\ldots\ldots$$

- At each time step t, algorithm generates a string z_t
- Uniformly generates in the limit if the moment the algorithm sees $t^\star = t^\star(\mathcal{C})$ distinct strings, all strings generated thereafter are new and in L_z

chooses L_{80}

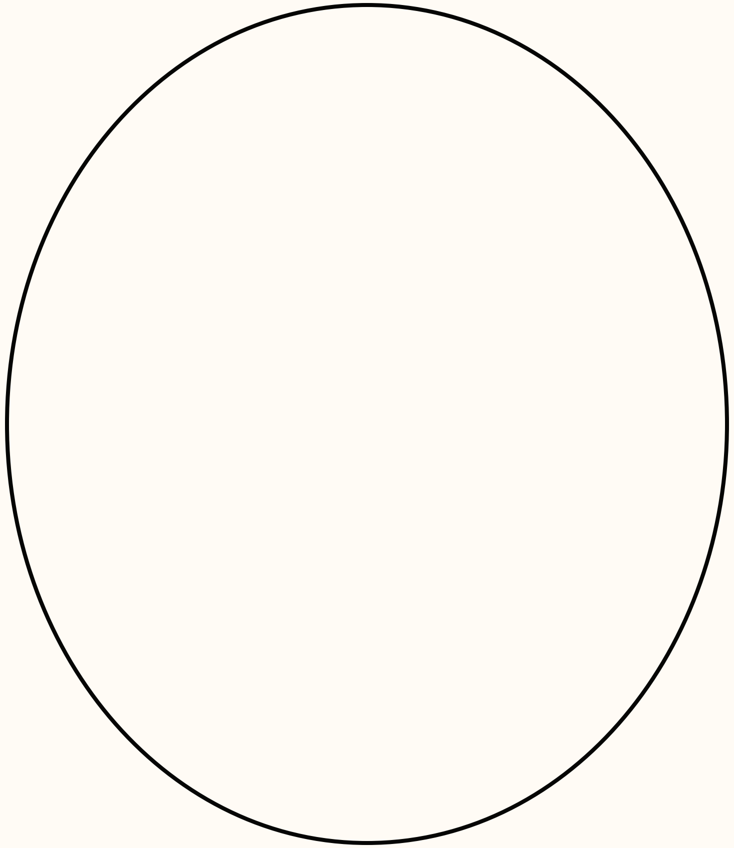
x'_1 x'_2 x'_3 x'_4 .... $x'_{t^\star}$ $x'_{t^\star+1}$ ....

z'_1 z'_2 z'_3 z'_4 .... $z'_{t^\star}$ $z'_{t^\star+1}$ ....

new, $\in L_{80}$ new, $\in L_{80}$

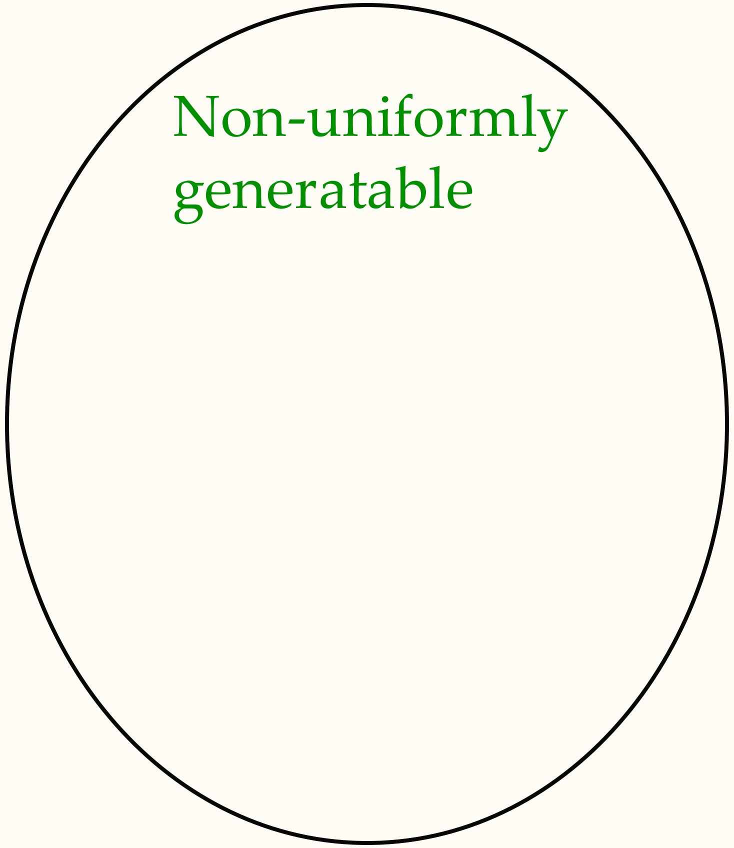
Uniform Generation in the Limit

Uniform Generation in the Limit



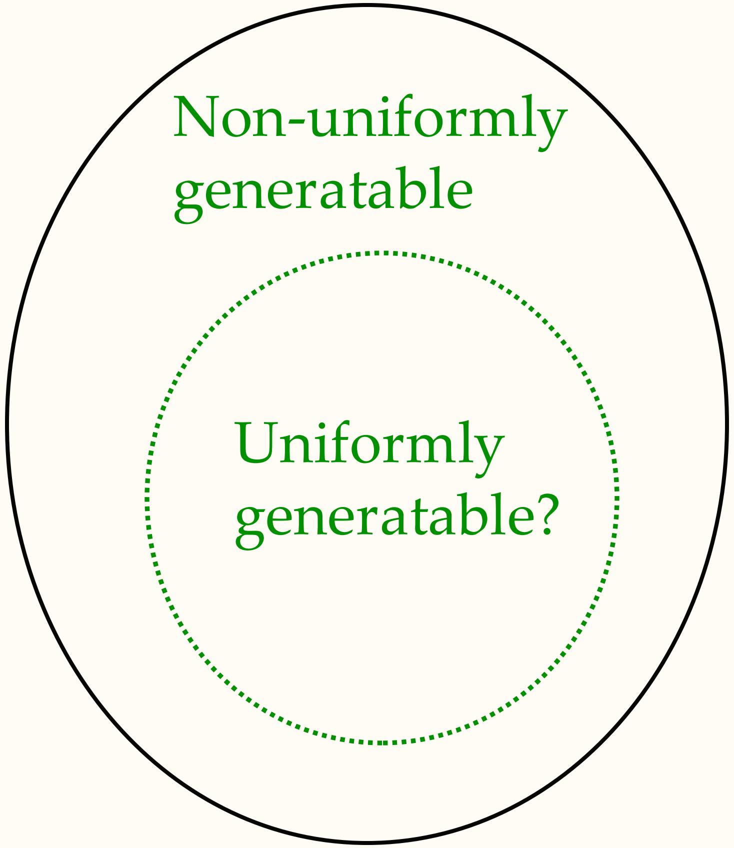
Countable language collections

Uniform Generation in the Limit



Countable language collections

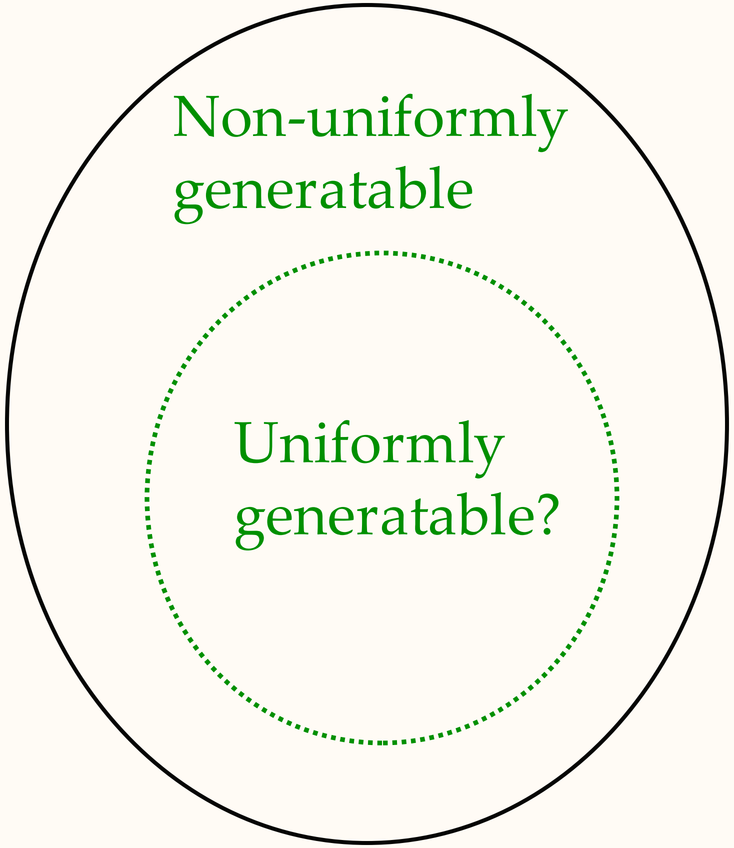
Uniform Generation in the Limit



Countable language collections

Uniform Generation in the Limit

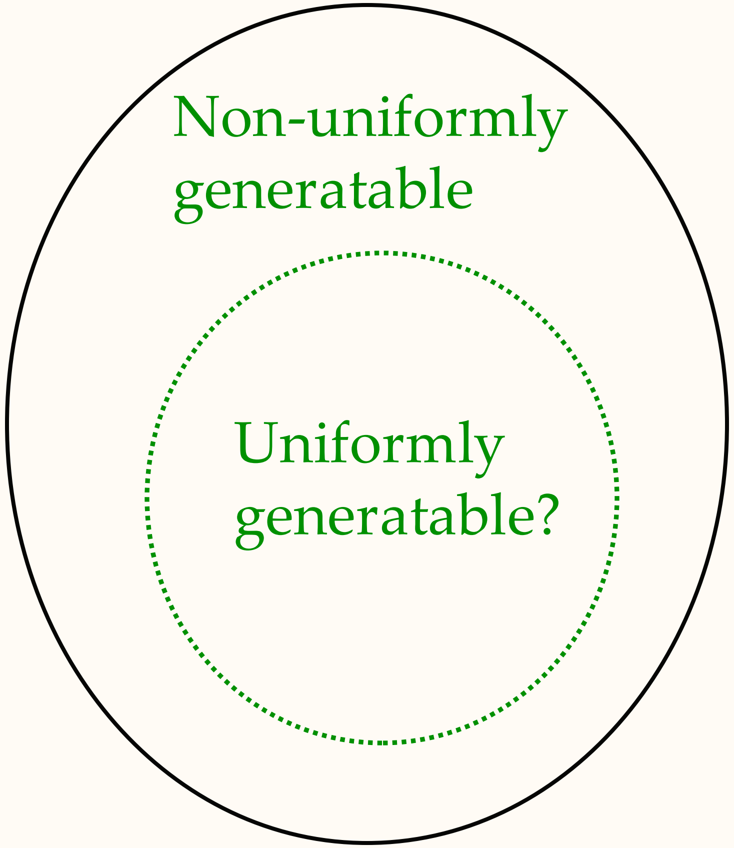
- $O = \{-1, -3, -5, \ldots\}$



Countable language collections

Uniform Generation in the Limit

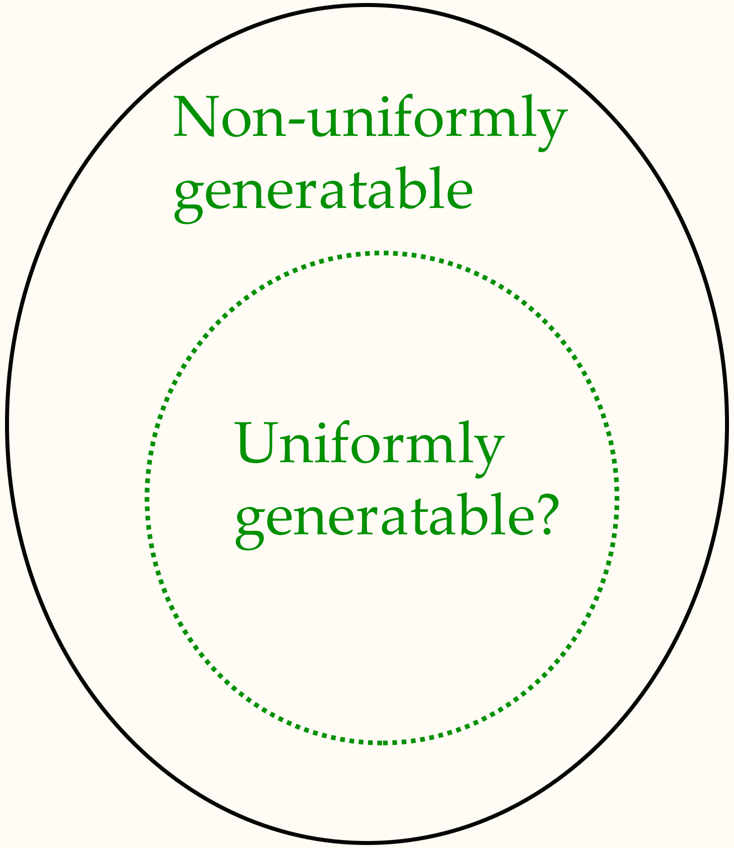
- $O = \{-1, -3, -5, \dots\}$
- $E = \{-2, -4, -6, \dots\}$



Countable language collections

Uniform Generation in the Limit

- $O = \{-1, -3, -5, \dots\}$
- $E = \{-2, -4, -6, \dots\}$
- Consider collection $\mathcal{C}$ that contains:

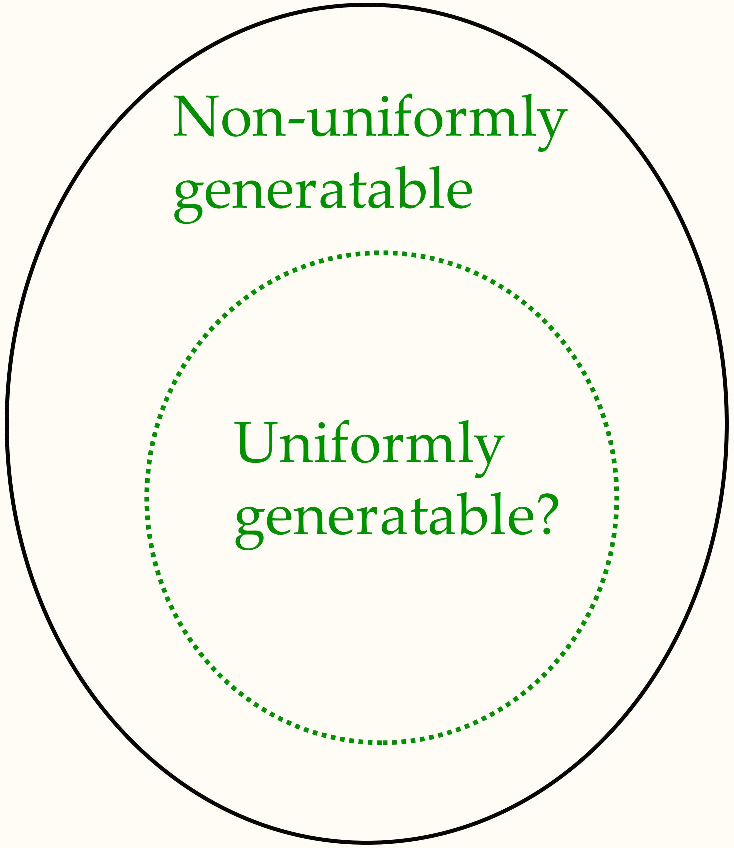


Countable language collections

Uniform Generation in the Limit

- $O = \{-1, -3, -5, \dots\}$
- $E = \{-2, -4, -6, \dots\}$
- Consider collection $\mathcal{C}$ that contains:

  $O_1 = O \cup \{1\}, E_1 = E \cup \{1\}$



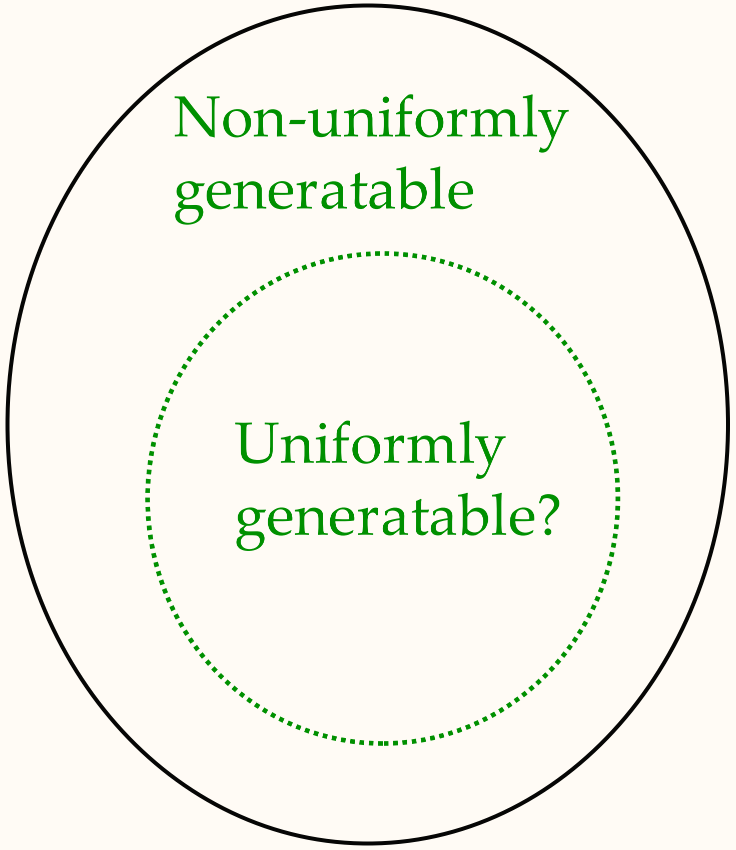
Countable language collections

Uniform Generation in the Limit

- $O = \{-1, -3, -5, \dots\}$
- $E = \{-2, -4, -6, \dots\}$
- Consider collection $\mathcal{C}$ that contains:

$$O_1 = O \cup \{1\}, E_1 = E \cup \{1\}$$

$$O_2 = O \cup \{1, 2\}, E_2 = E \cup \{1, 2\}$$



Countable language collections

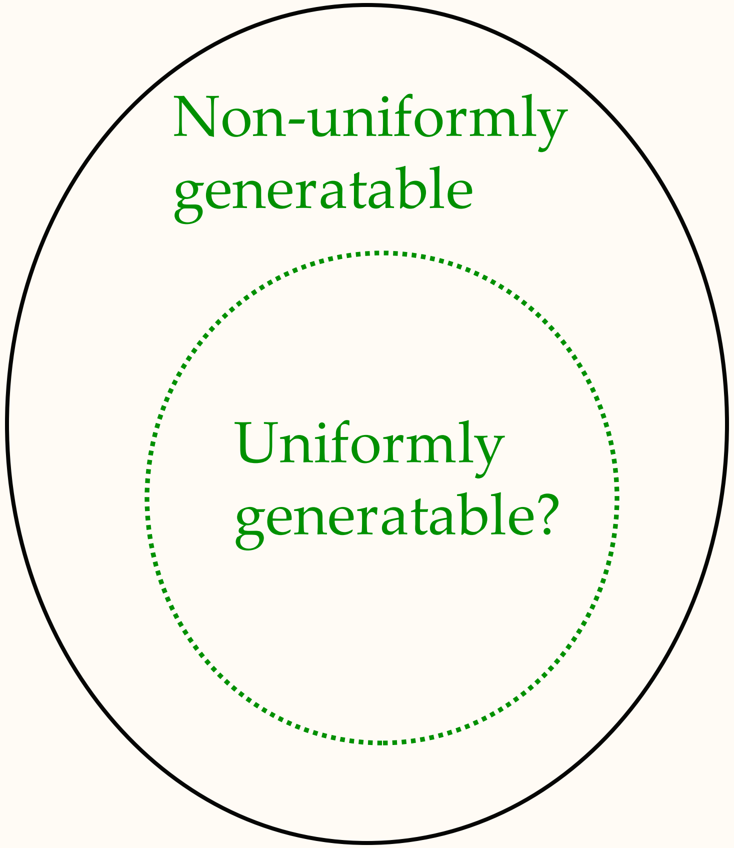
Uniform Generation in the Limit

- $O = \{-1, -3, -5, \dots\}$
- $E = \{-2, -4, -6, \dots\}$
- Consider collection $\mathcal{C}$ that contains:

$$O_1 = O \cup \{1\}, E_1 = E \cup \{1\}$$

$$O_2 = O \cup \{1, 2\}, E_2 = E \cup \{1, 2\}$$

$$\vdots$$



Countable language collections

Uniform Generation in the Limit

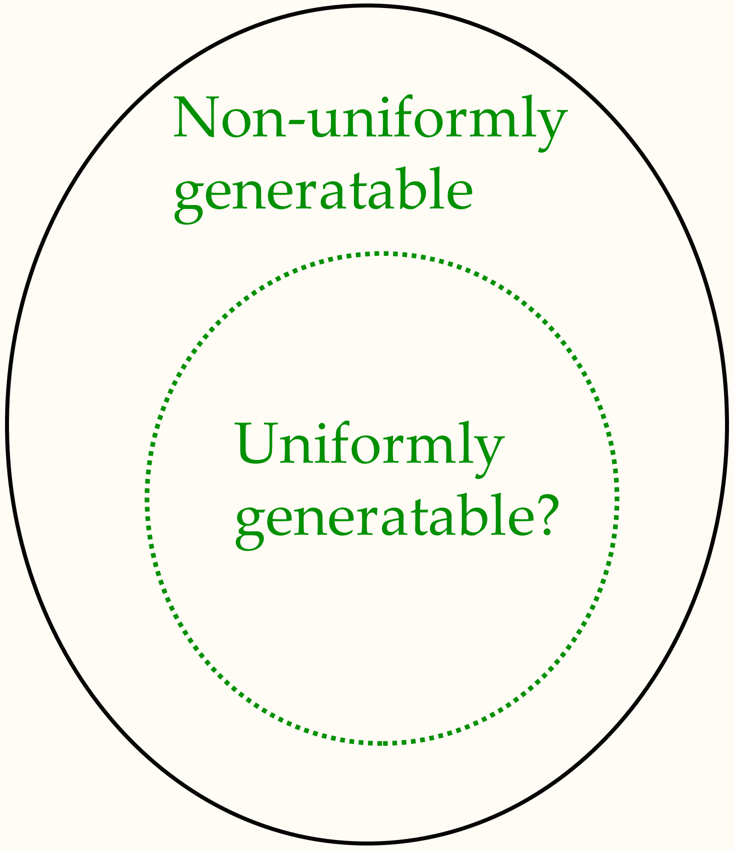
- $O = \{-1, -3, -5, \ldots\}$
- $E = \{-2, -4, -6, \ldots\}$
- Consider collection $\mathcal{C}$ that contains:

$$O_1 = O \cup \{1\}, E_1 = E \cup \{1\}$$

$$O_2 = O \cup \{1, 2\}, E_2 = E \cup \{1, 2\}$$

$$\vdots$$

$$O_n = O \cup \{1, 2, \ldots, n\}, E_n = E \cup \{1, 2, \ldots, n\}$$



Countable language collections

Uniform Generation in the Limit

- $O = \{-1, -3, -5, \dots\}$
- $E = \{-2, -4, -6, \dots\}$
- Consider collection $\mathcal{C}$ that contains:

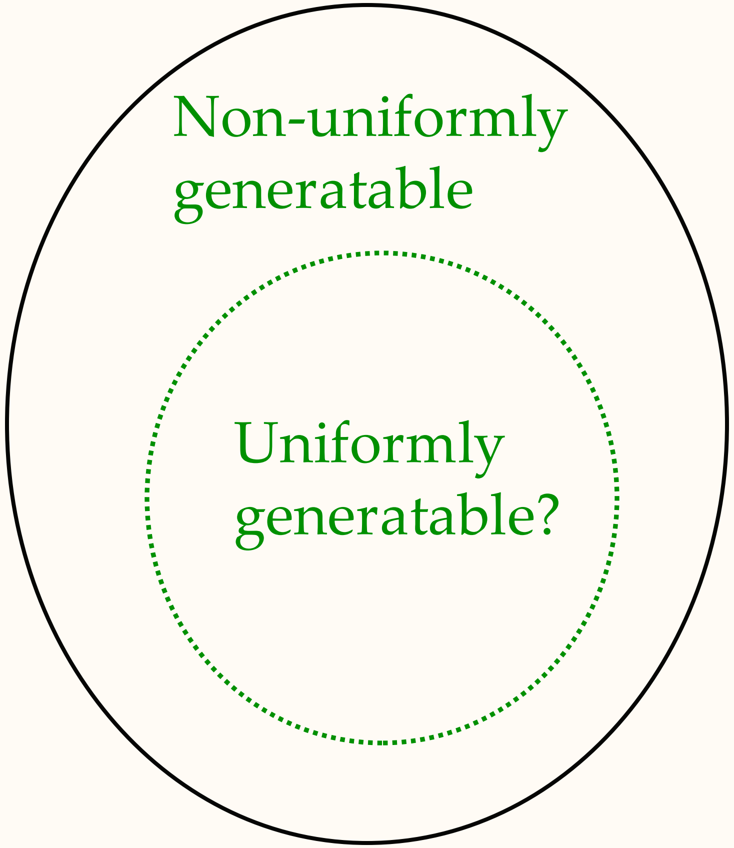
$$O_1 = O \cup \{1\}, E_1 = E \cup \{1\}$$

$$O_2 = O \cup \{1, 2\}, E_2 = E \cup \{1, 2\}$$

$$\vdots$$

$$O_n = O \cup \{1, 2, \dots, n\}, E_n = E \cup \{1, 2, \dots, n\}$$

$$\vdots$$



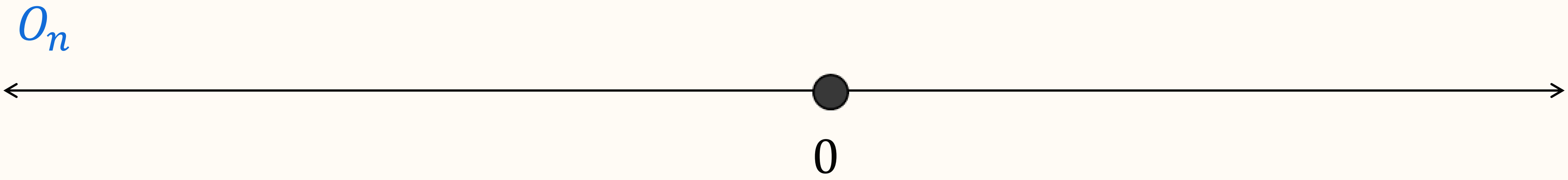
Countable language collections

Uniform Generation in the Limit

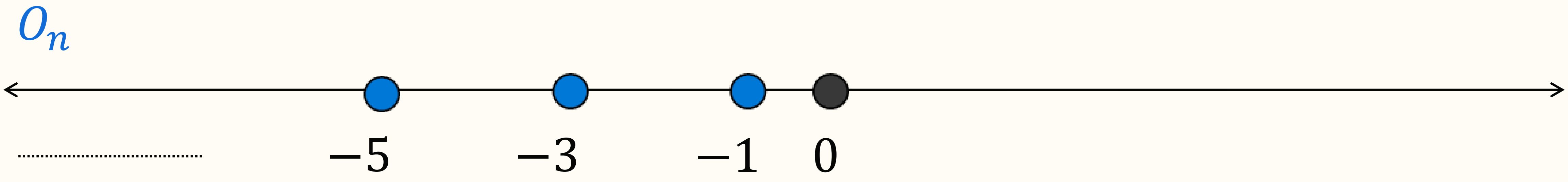
Uniform Generation in the Limit

o_n

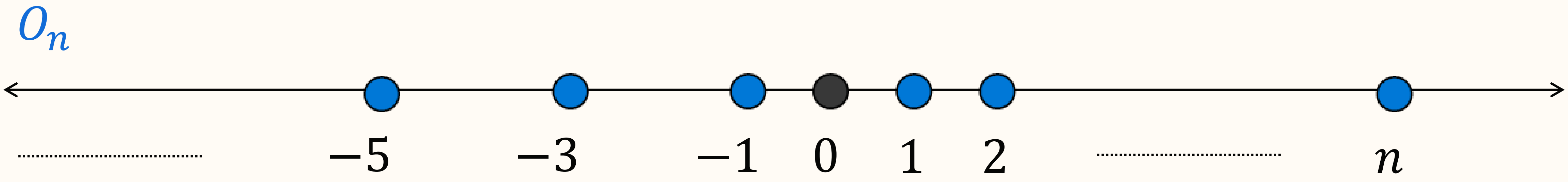
Uniform Generation in the Limit



Uniform Generation in the Limit

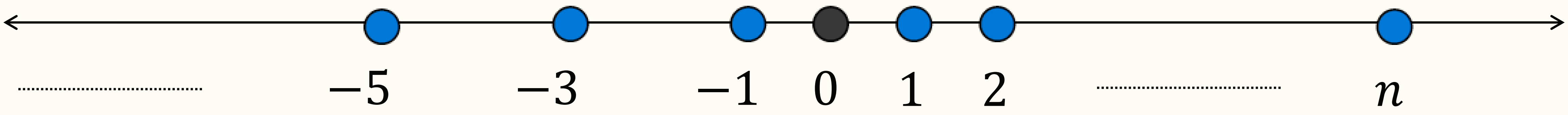


Uniform Generation in the Limit



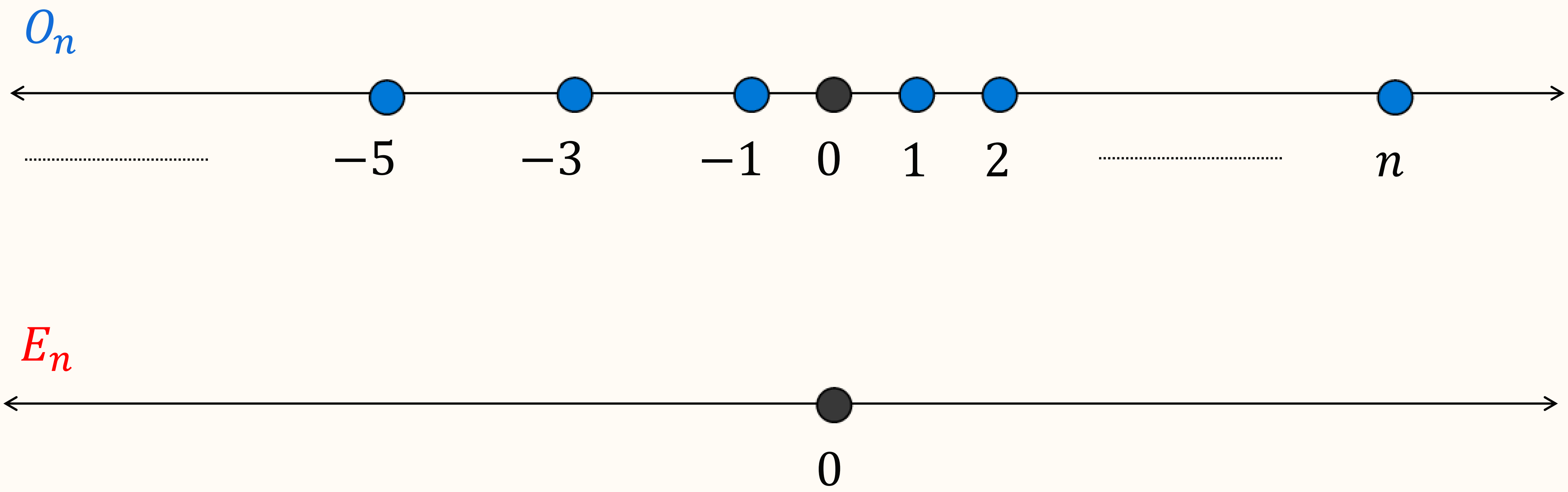
Uniform Generation in the Limit

O_n

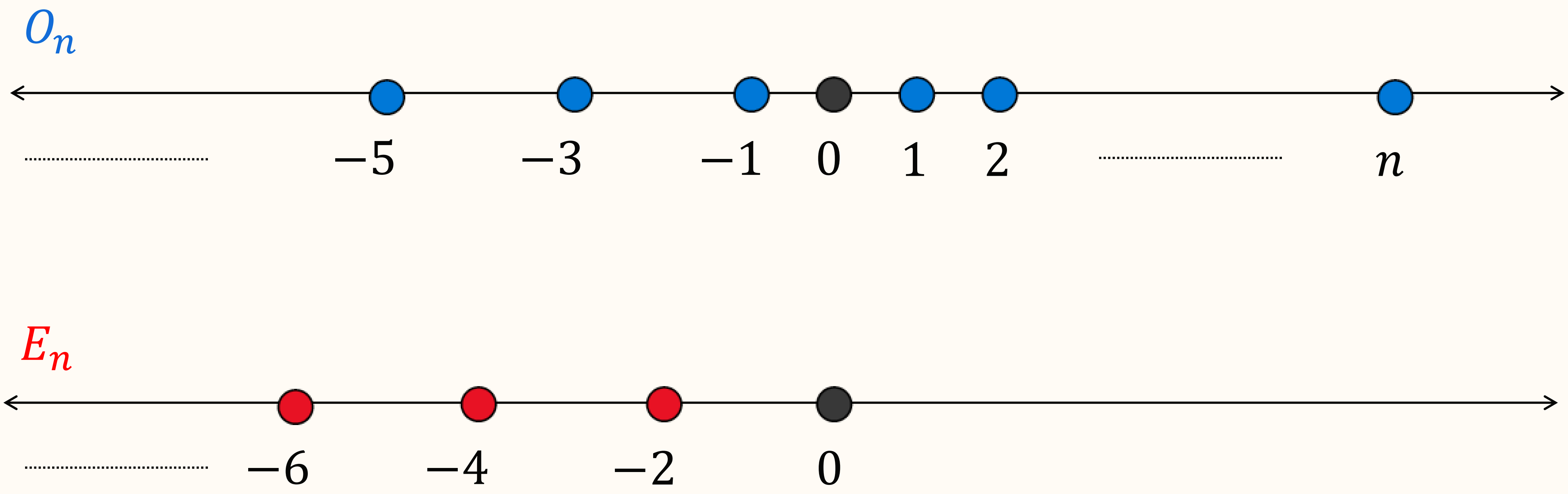


E_n

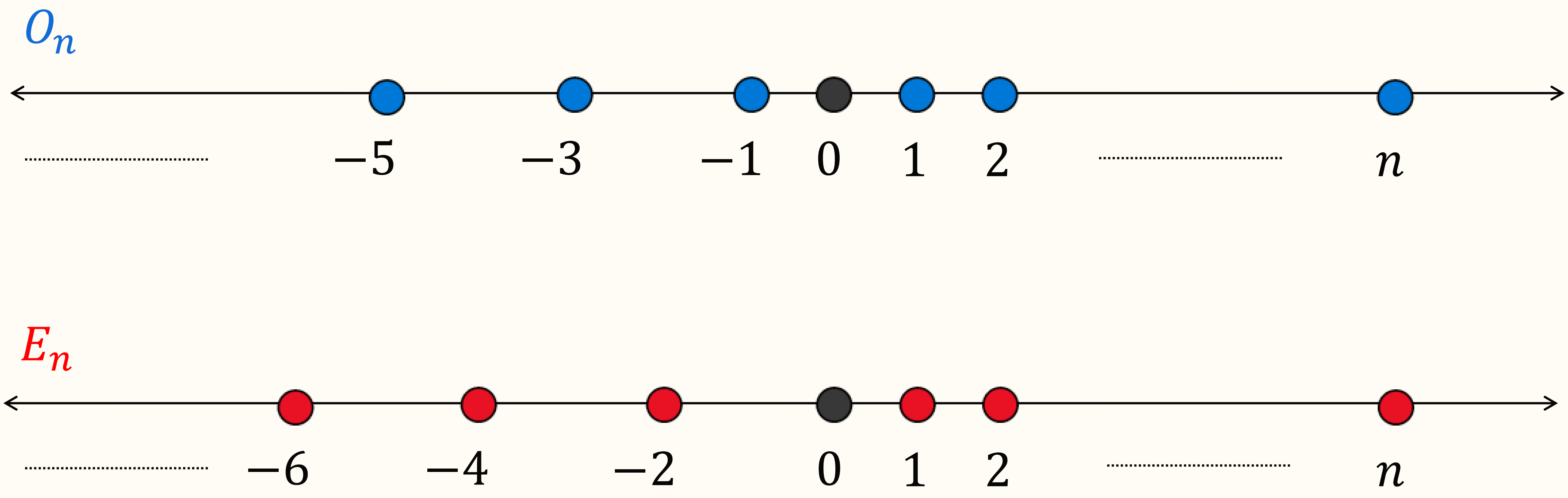
Uniform Generation in the Limit



Uniform Generation in the Limit



Uniform Generation in the Limit



Uniform Generation in the Limit

Uniform Generation in the Limit

- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$

Uniform Generation in the Limit

- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \ldots, t^\star\}$

Uniform Generation in the Limit

- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \ldots, t^\star\}$

Uniform Generation in the Limit

- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \ldots, t^\star\}$

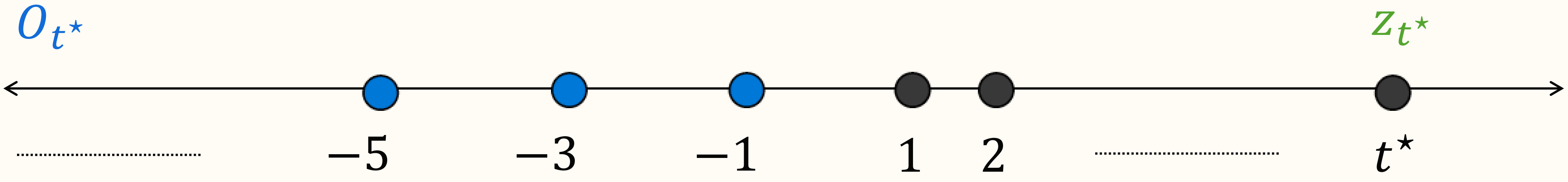
Uniform Generation in the Limit

- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \ldots, t^\star\}$

- If $z_{t^\star} \in E$, adversary continues enumerating O

Uniform Generation in the Limit

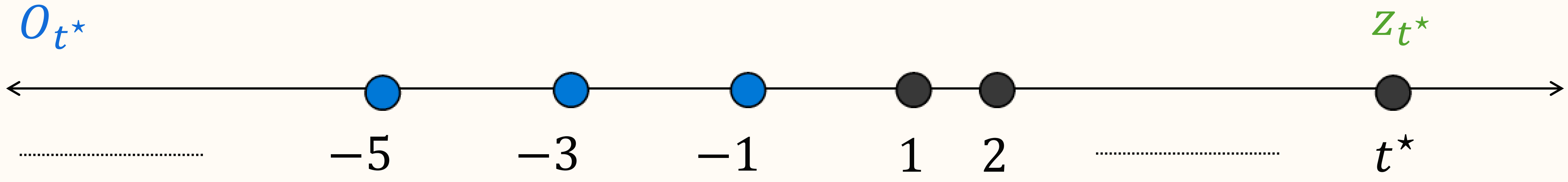
- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \ldots, t^\star\}$



- If $z_{t^\star} \in E$, adversary continues enumerating O

Uniform Generation in the Limit

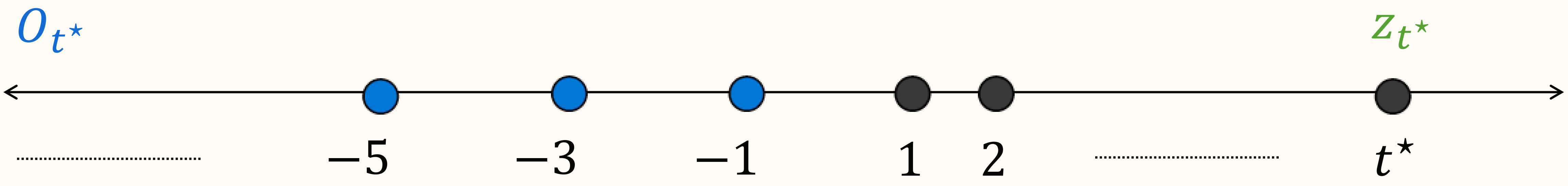
- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \dots, t^\star\}$



- If $z_{t^\star} \in E$, adversary continues enumerating O

Uniform Generation in the Limit

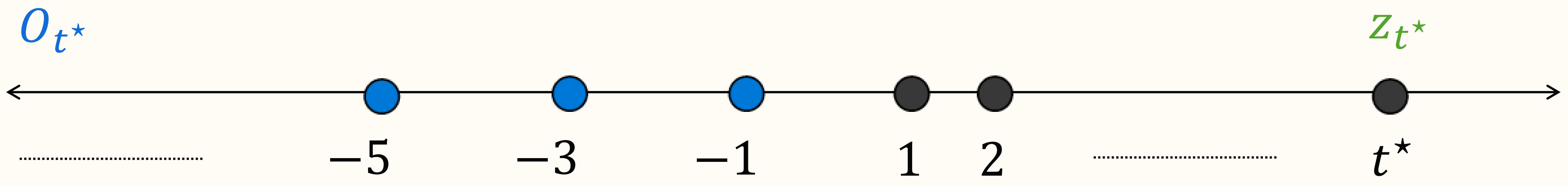
- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \dots, t^\star\}$



- If $z_{t^\star} \in E$, adversary continues enumerating O

Uniform Generation in the Limit

- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \ldots, t^\star\}$

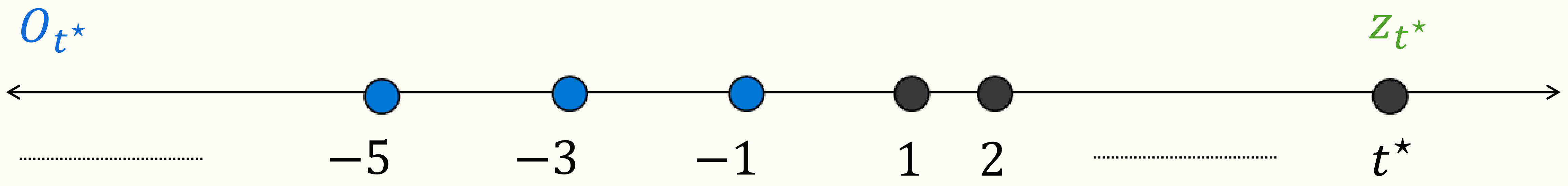


- If $z_{t^\star} \in E$, adversary continues enumerating O

- If $z_{t^\star} \in O$, adversary continues enumerating E

Uniform Generation in the Limit

- Suppose algorithm claims a uniform time bound of $t^\star = t^\star(\mathcal{C})$
- Adversary enumerates $\{1, 2, \dots, t^\star\}$

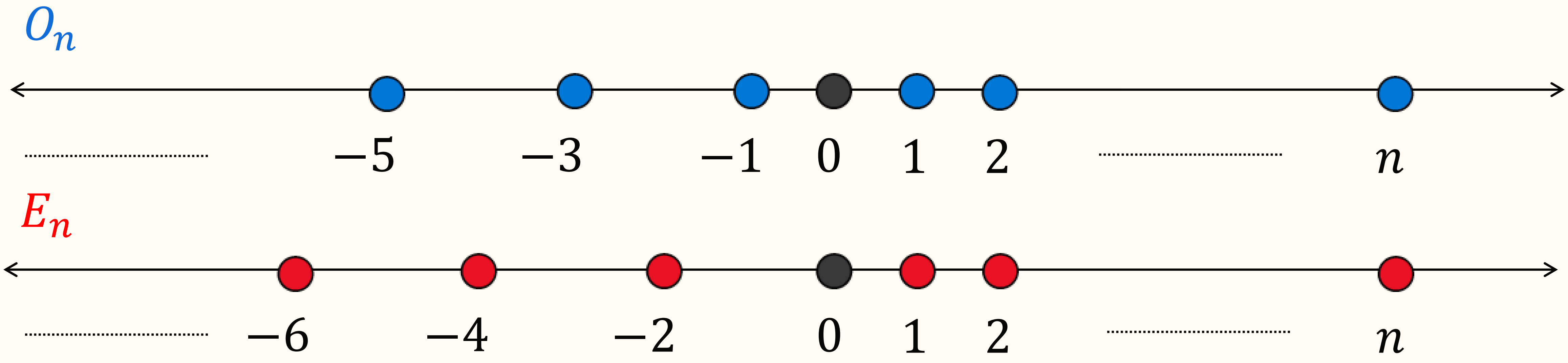


- If $z_{t^\star} \in E$, adversary continues enumerating O

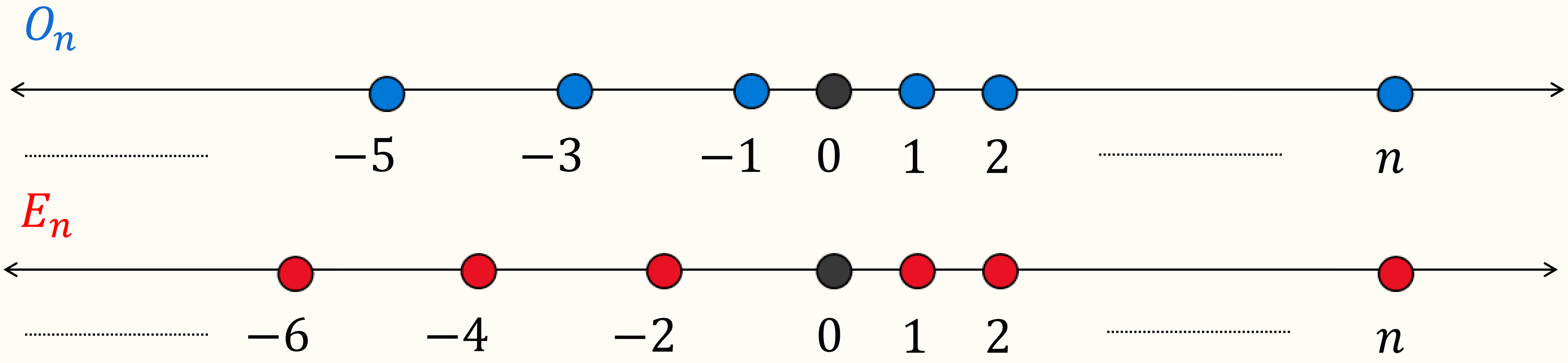
- If $z_{t^\star} \in O$, adversary continues enumerating E

Uniform Generation in the Limit

Uniform Generation in the Limit

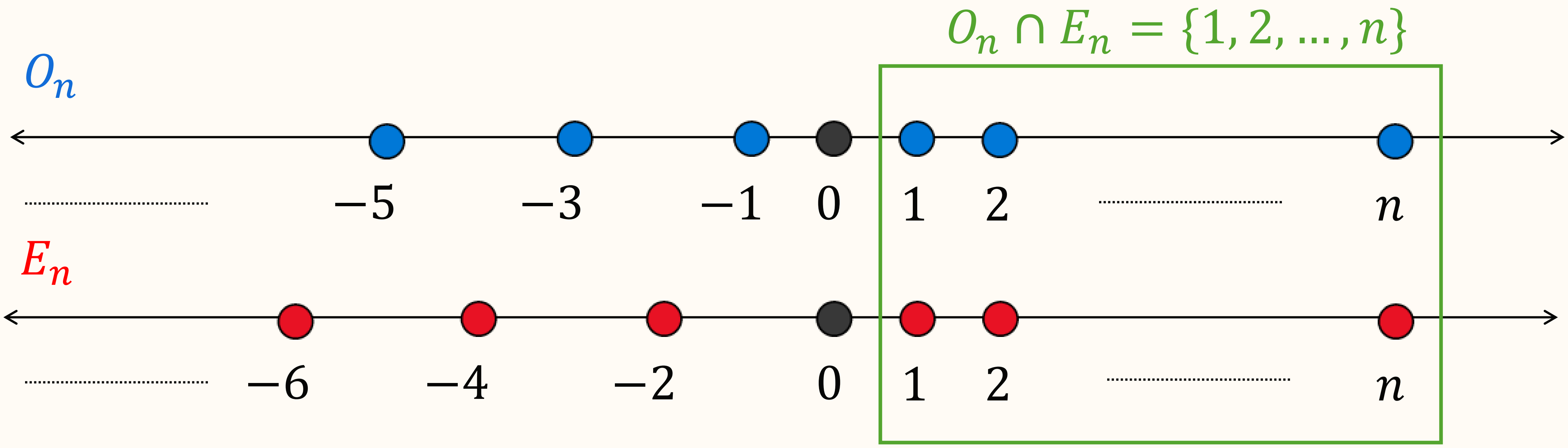


Uniform Generation in the Limit



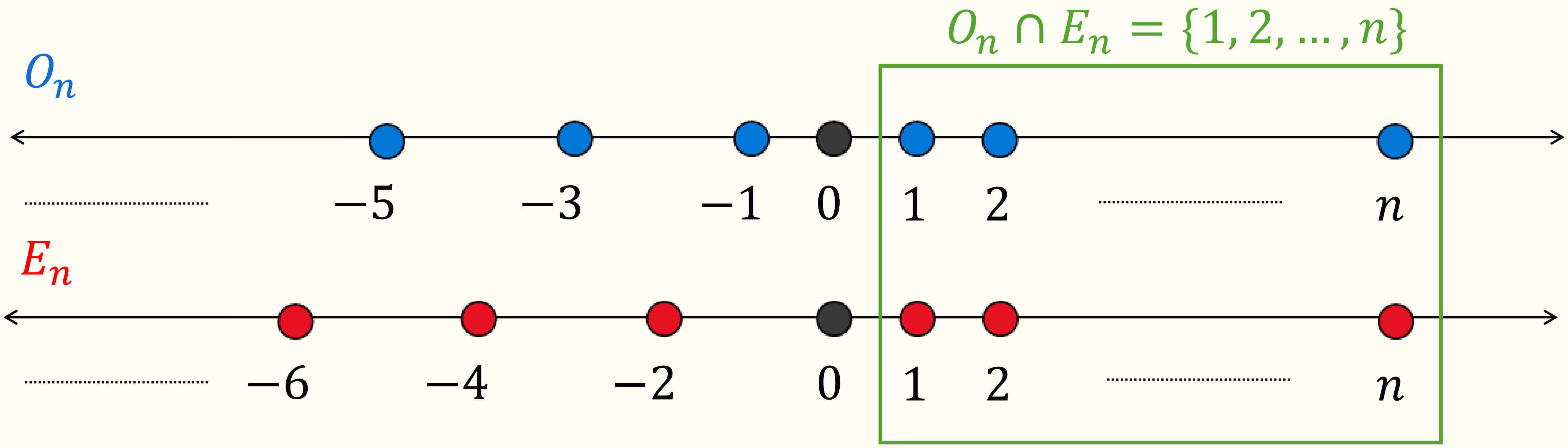
- Issue: Arbitrarily large **finite intersections**

Uniform Generation in the Limit



- Issue: Arbitrarily large **finite intersections**

Uniform Generation in the Limit



- Issue: Arbitrarily large **finite intersections**
- <u>Definition:</u> Largest finite intersection $:=$ "closure dimension"

Uniform Generation in the Limit

Uniform Generation in the Limit

- Theorem (Kleinberg, Mullainathan ‘24, Li, Raman, Tewari ‘24): Uniformly generatable if and only closure dimension is bounded

Uniform Generation in the Limit

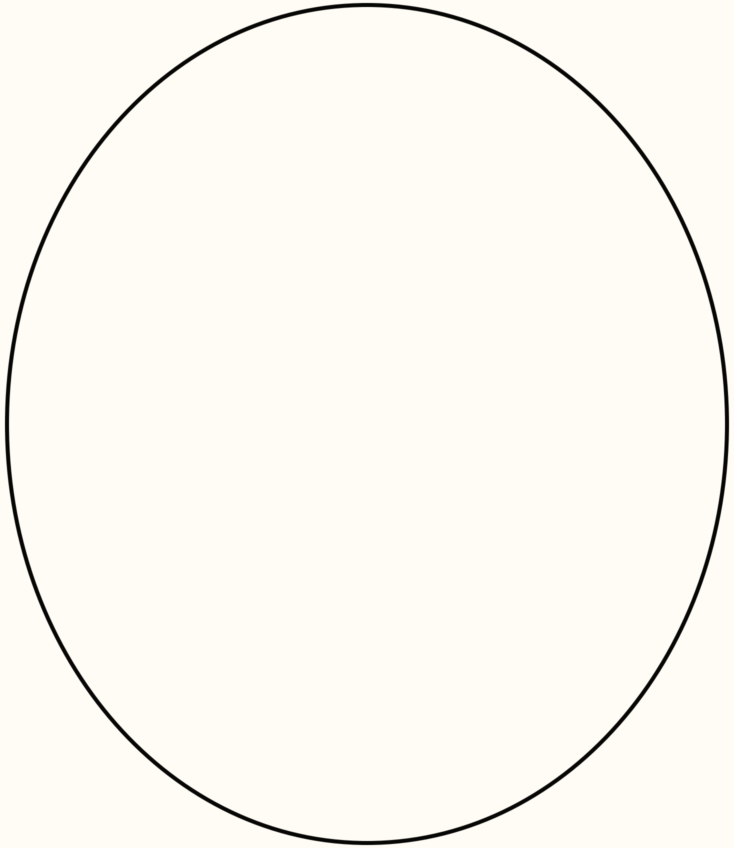
- Theorem (Kleinberg, Mullainathan '24, Li, Raman, Tewari '24): Uniformly generatable if and only closure dimension is bounded

- $t^\star(\mathcal{C})$ = closure dimension

Uniform Generation in the Limit

- Theorem (Kleinberg, Mullainathan '24, Li, Raman, Tewari '24): Uniformly generatable if and only closure dimension is bounded

- $t^{\star}(\mathcal{C})$ = closure dimension

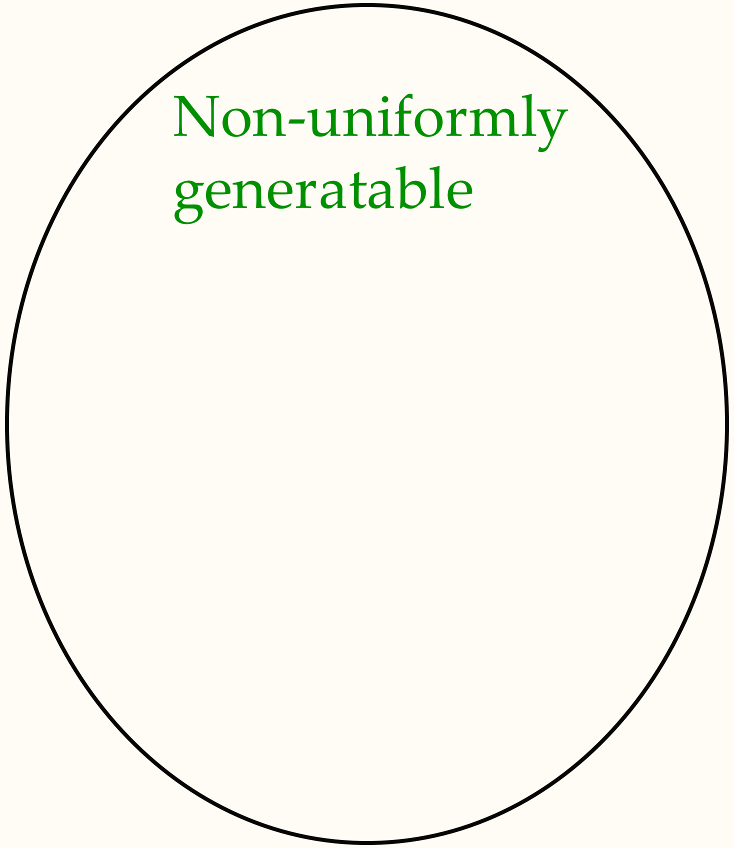


Countable language collections

Uniform Generation in the Limit

- Theorem (Kleinberg, Mullainathan ‘24, Li, Raman, Tewari ‘24): Uniformly generatable if and only closure dimension is bounded

- $t^{\star}(\mathcal{C})$ = closure dimension

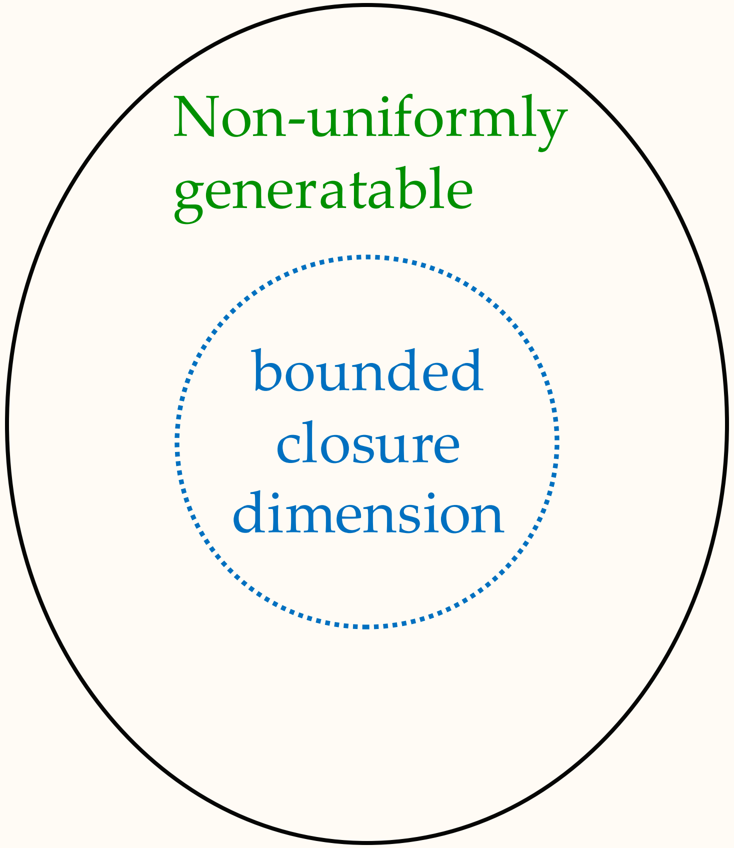


Countable language collections

Uniform Generation in the Limit

- Theorem (Kleinberg, Mullainathan '24, Li, Raman, Tewari '24): Uniformly generatable if and only closure dimension is bounded

- $t^{\star}(\mathcal{C})$ = closure dimension

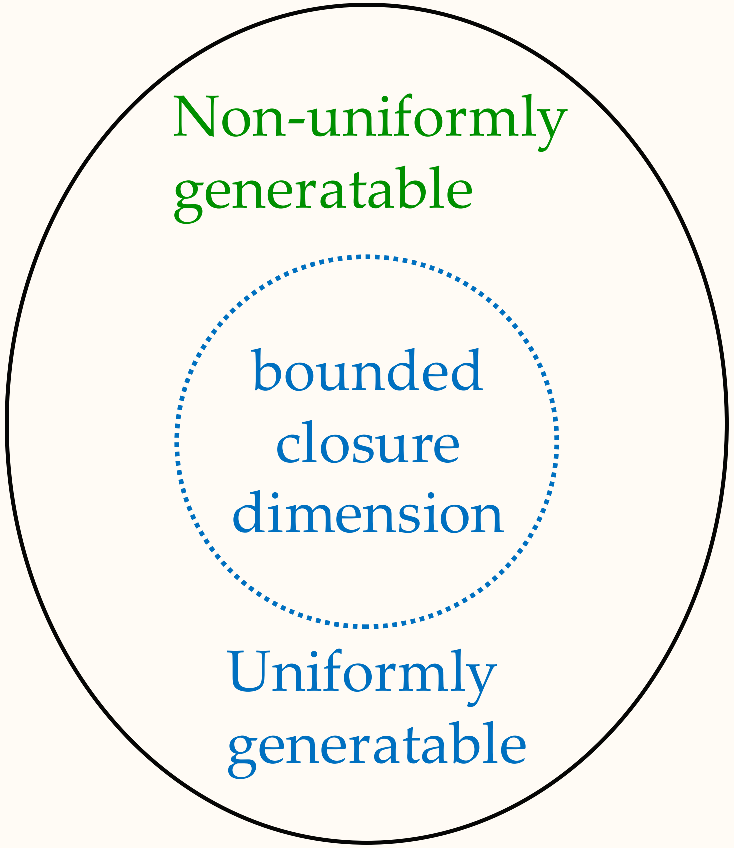


Countable language collections

Uniform Generation in the Limit

- Theorem (Kleinberg, Mullainathan '24, Li, Raman, Tewari '24): Uniformly generatable if and only closure dimension is bounded

- $t^\star(\mathcal{C})$ = closure dimension



Countable language collections

Languages as binary concepts

Languages as binary concepts

Associate every language with its indicator function

Languages as binary concepts

Associate every language with its indicator function

$$\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$$

$$L_1 = \{x_1, x_2, x_5, \ldots\}$$

$$L_2 = \{x_4, x_5, \ldots\}$$

$$L_3 = \{x_1, x_3, x_4, \ldots\}$$

$$\vdots$$

Languages as binary concepts

Associate every language with its indicator function

$$\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$$

$$L_1 = \{x_1, x_2, x_5, \ldots\}$$

$$L_2 = \{x_4, x_5, \ldots\}$$

$$L_3 = \{x_1, x_3, x_4, \ldots\}$$

$$\vdots$$

$\mathcal{H}$ \ $\mathcal{X}$	x_1	x_2	x_3	x_4	x_5	…
h_1	1	1	0	0	1	
h_2	0	0	0	1	1	
h_3	1	0	1	1	0	
$\vdots$						

Languages as binary concepts

Associate every language with its indicator function

$$\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$$

$$L_1 = \{x_1, x_2, x_5, \ldots\}$$

$$L_2 = \{x_4, x_5, \ldots\}$$

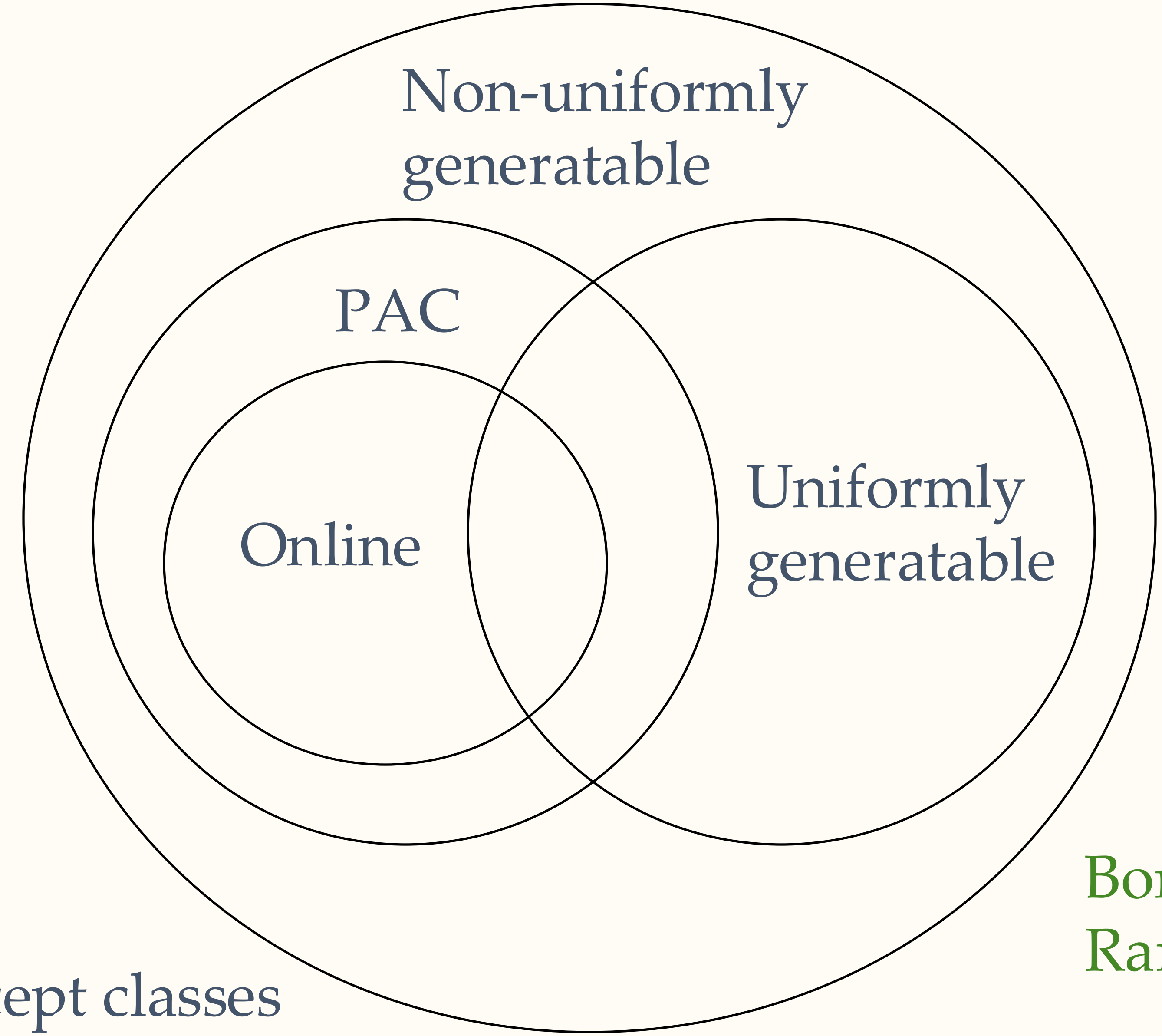
$$L_3 = \{x_1, x_3, x_4, \ldots\}$$

$$\vdots$$

$\mathcal{H}$ \ $\mathcal{X}$	x_1	x_2	x_3	x_4	x_5	$\cdots$
h_1	1	1	0	0	1	
h_2	0	0	0	1	1	
h_3	1	0	1	1	0	
$\vdots$						

Can then ask standard learning theory questions for this concept class

Generation versus Prediction

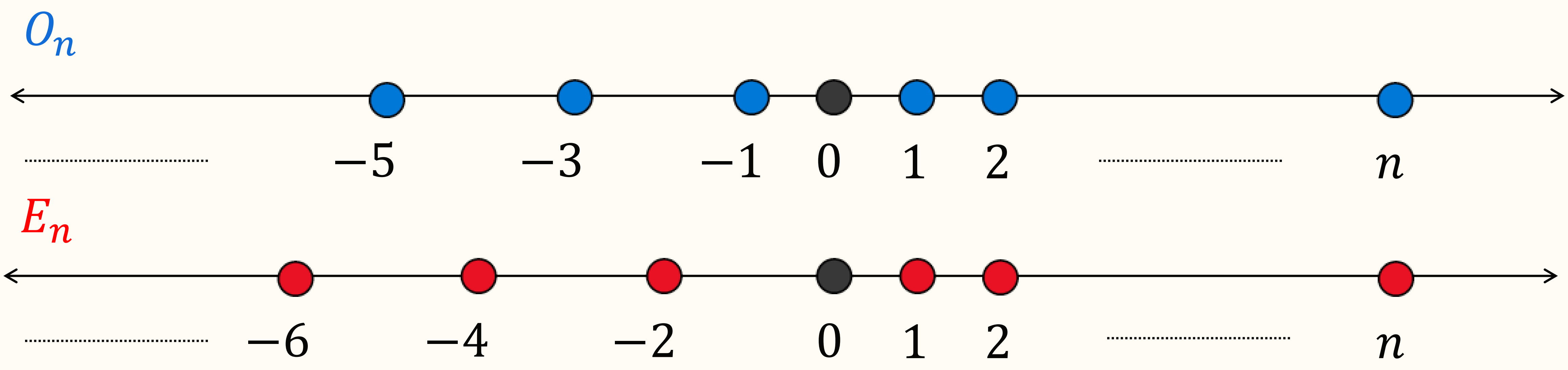


Borrowed from Li, Raman, Tewari '24

Countable concept classes

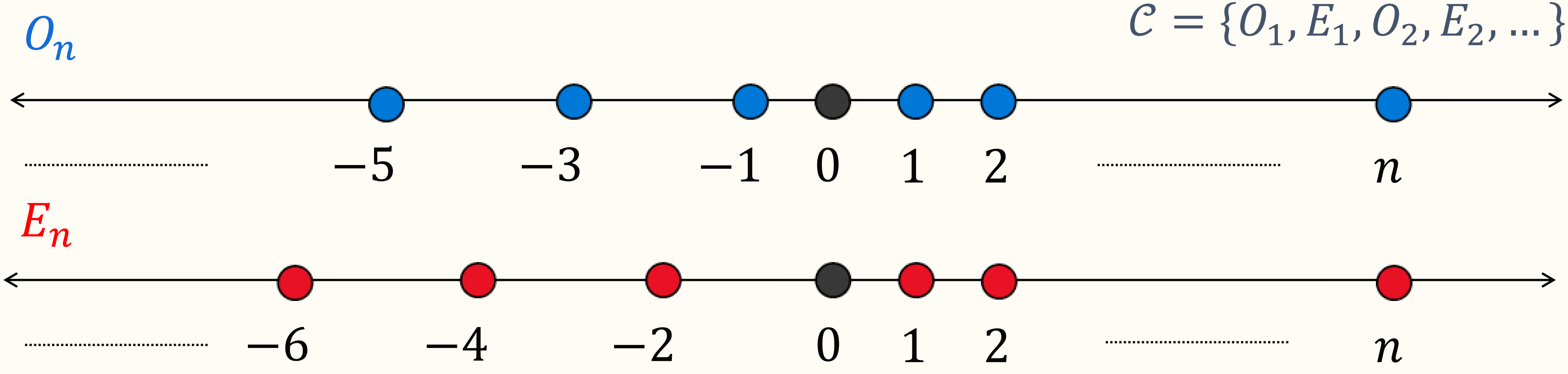
Generation versus Prediction

Generation versus Prediction



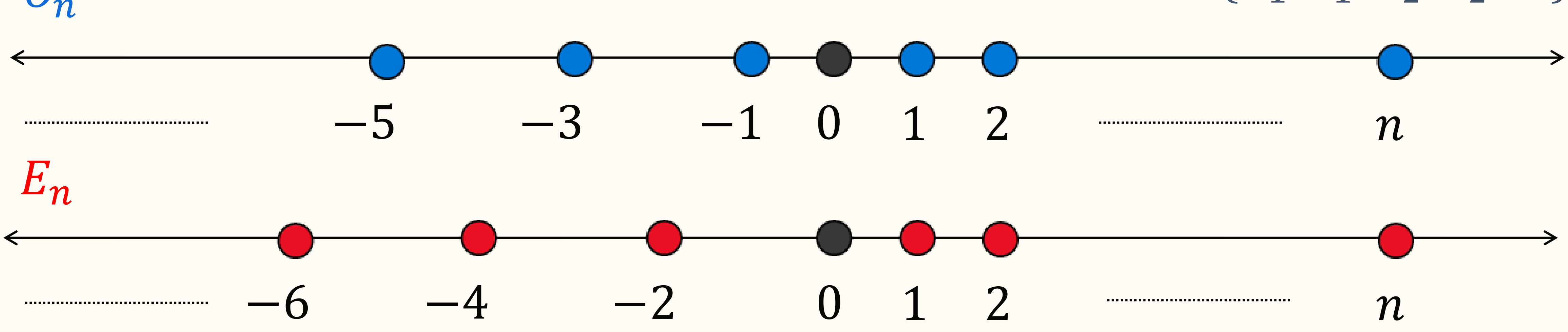
Generation versus Prediction

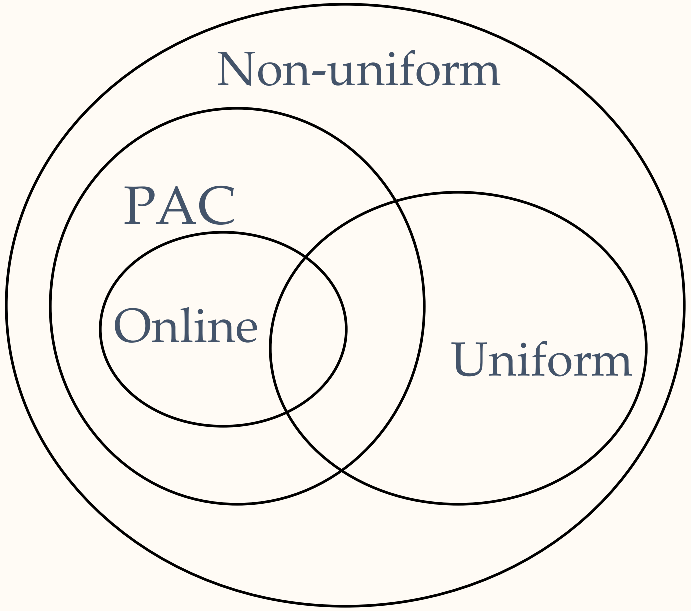
$\mathcal{C} = \{O_1, E_1, O_2, E_2, \ldots\}$



Generation versus Prediction

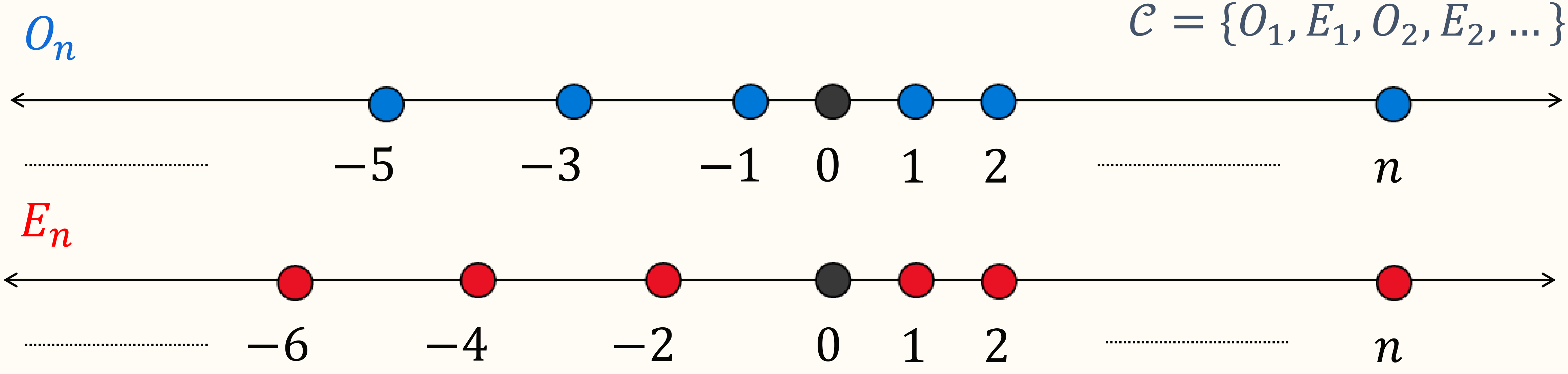
$\mathcal{C} = \{O_1, E_1, O_2, E_2, \ldots\}$



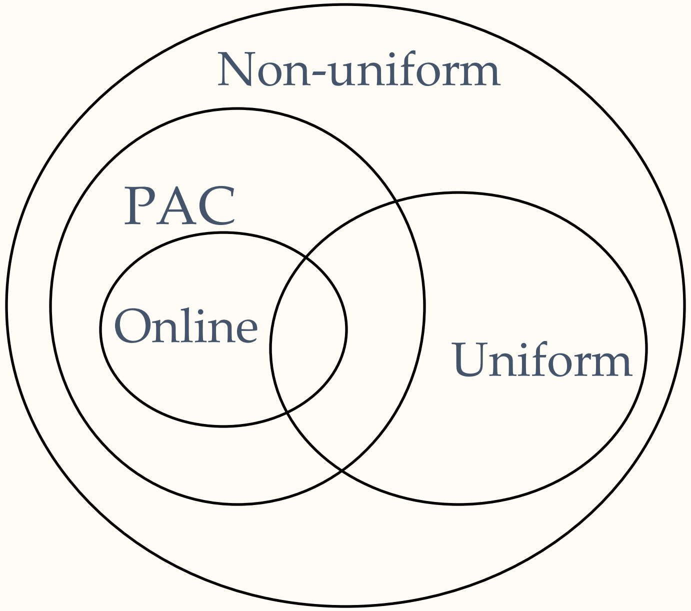


Generation versus Prediction

$\mathcal{C} = \{O_1, E_1, O_2, E_2, \dots\}$

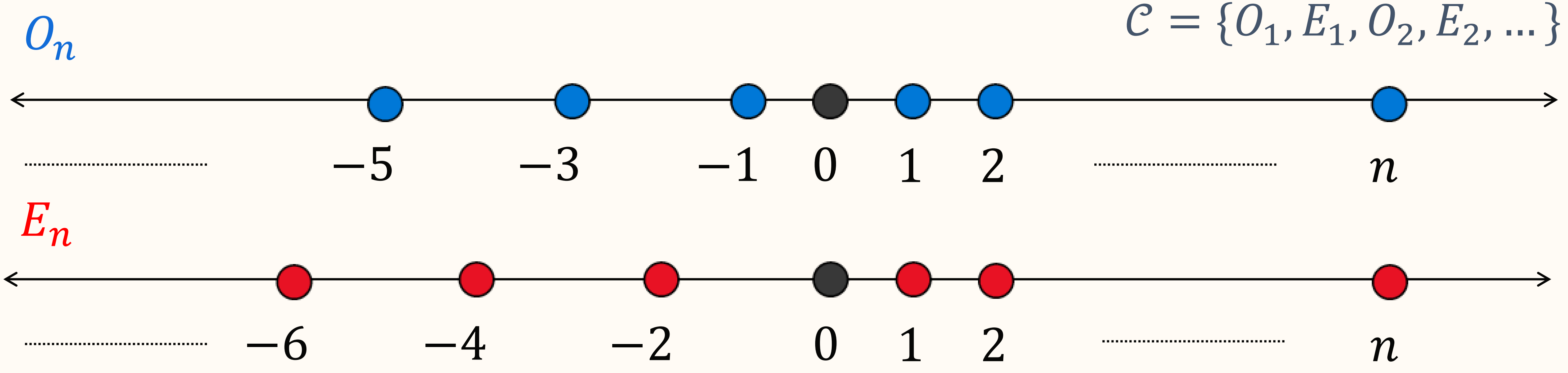


- Saw earlier that this is not uniformly generatable

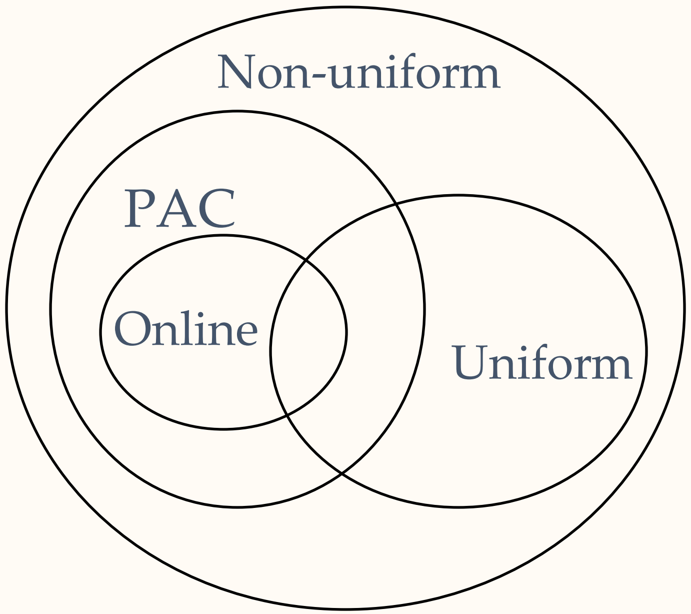


Generation versus Prediction

$\mathcal{C} = \{O_1, E_1, O_2, E_2, \ldots\}$

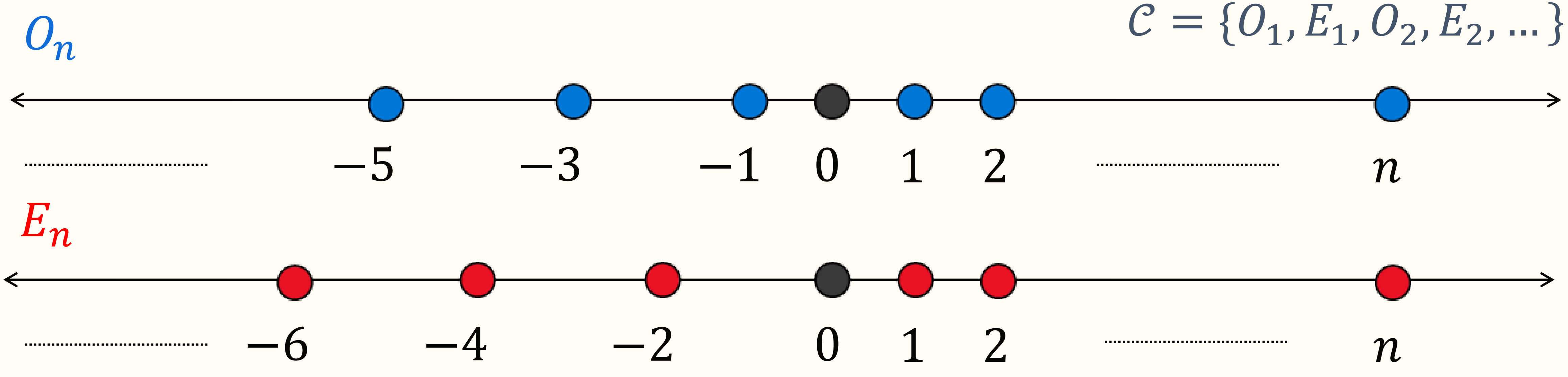


- Saw earlier that this is not uniformly generatable
- Not too hard to see that VC dimension is some small constant $\Longrightarrow$ PAC learnable

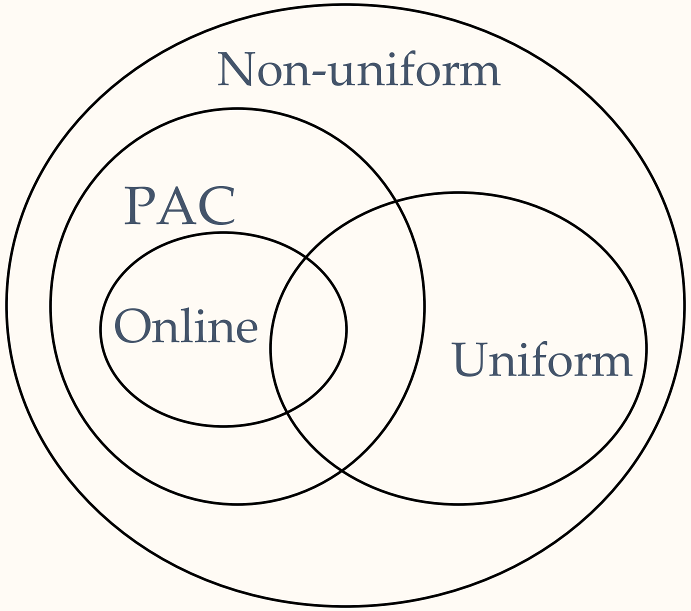


Generation versus Prediction

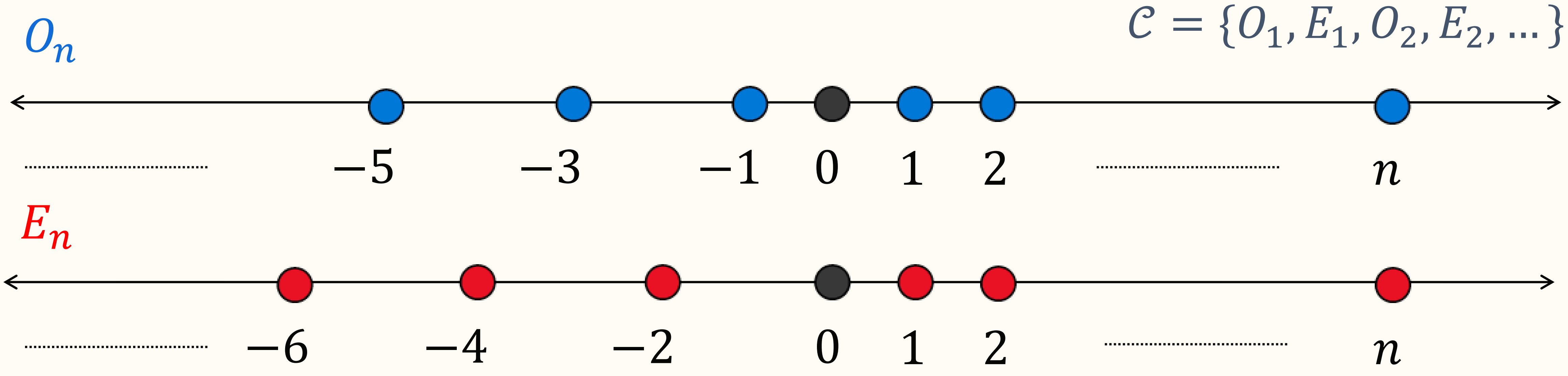
$\mathcal{C} = \{O_1, E_1, O_2, E_2, \dots\}$



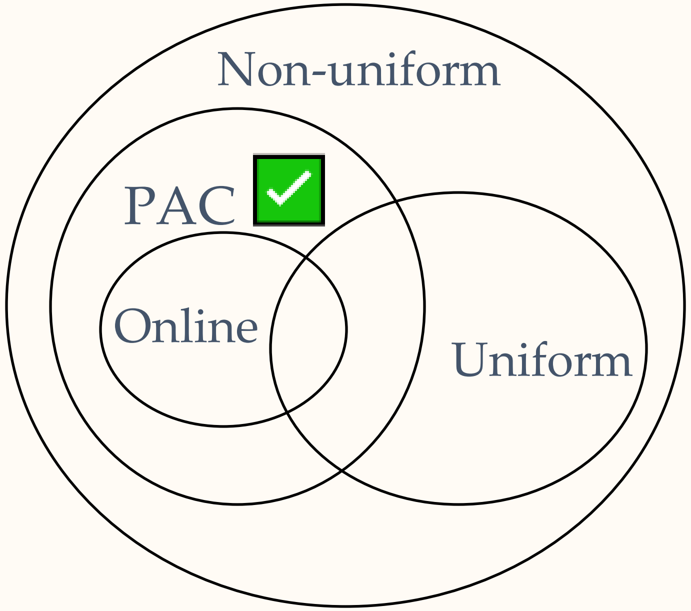
- Saw earlier that this is not uniformly generatable
- Not too hard to see that VC dimension is some small constant $\Longrightarrow$ PAC learnable
- Restriction on positive integers corresponds to thresholds $\Longrightarrow$ Not online learnable



Generation versus Prediction



- Saw earlier that this is not uniformly generatable
- Not too hard to see that VC dimension is some small constant $\Longrightarrow$ PAC learnable
- Restriction on positive integers corresponds to thresholds $\Longrightarrow$ Not online learnable

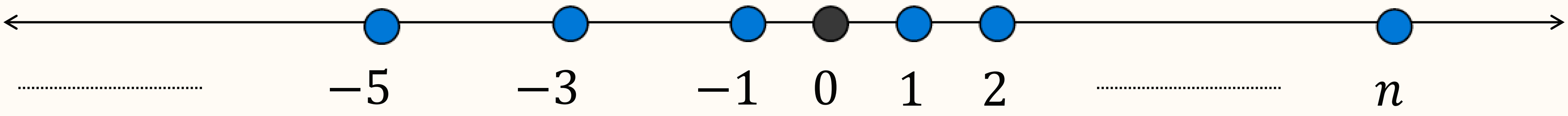


Generation versus Prediction

Generation versus Prediction

O_n

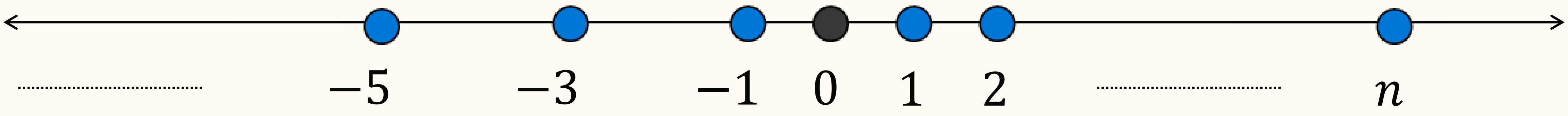
$\mathcal{C} = \{O_1, O_2, O_3, \dots\}$

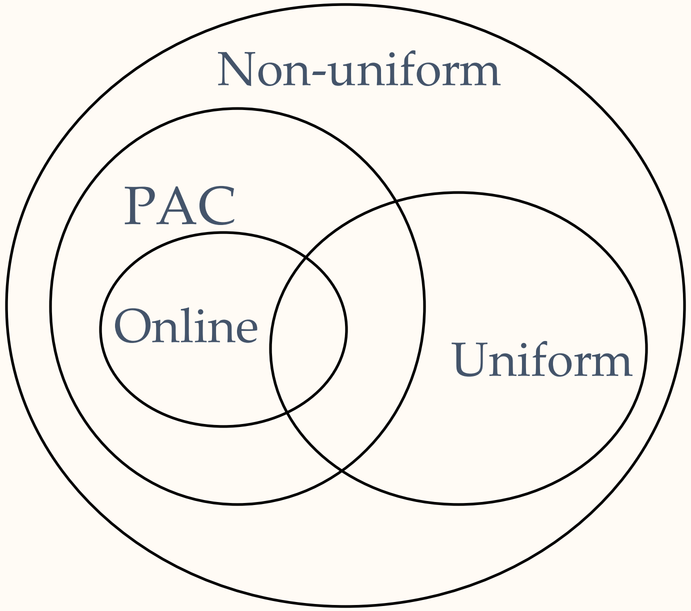


Generation versus Prediction

O_n

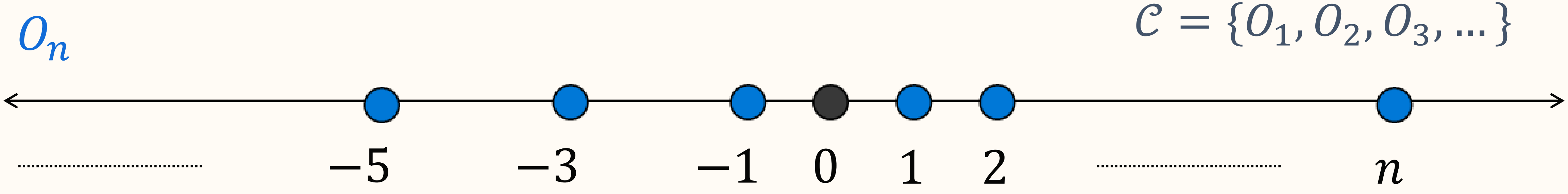
$\mathcal{C} = \{O_1, O_2, O_3, \dots\}$



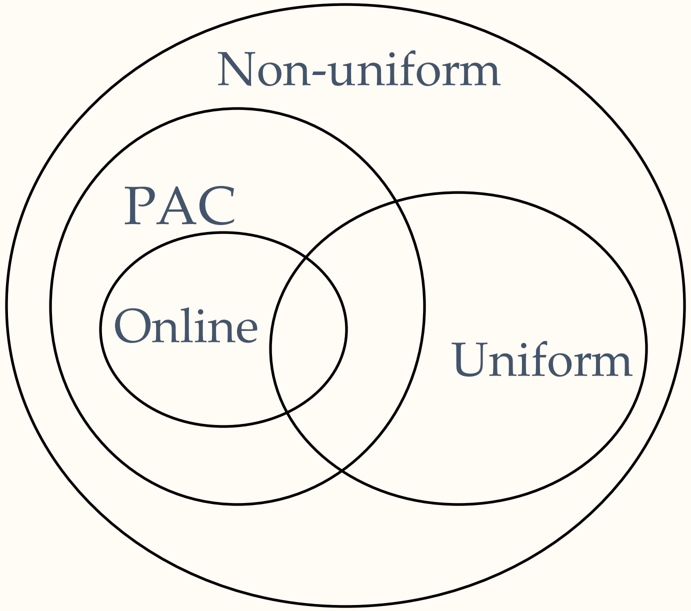


Generation versus Prediction

$\mathcal{C} = \{O_1, O_2, O_3, \dots\}$

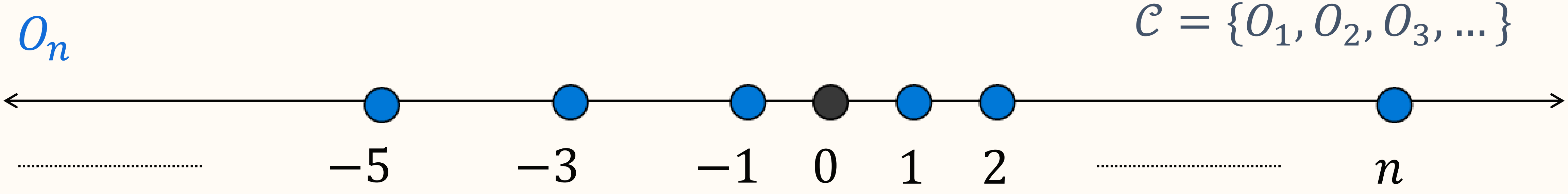


- Every language contains 0

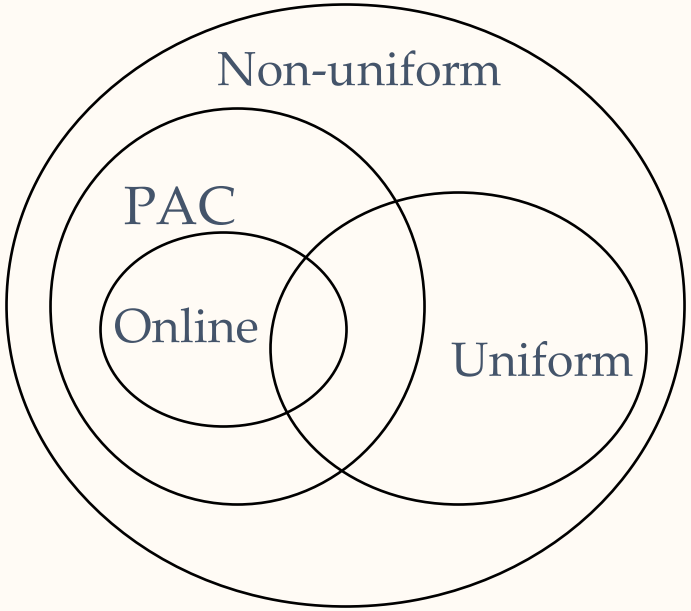


Generation versus Prediction

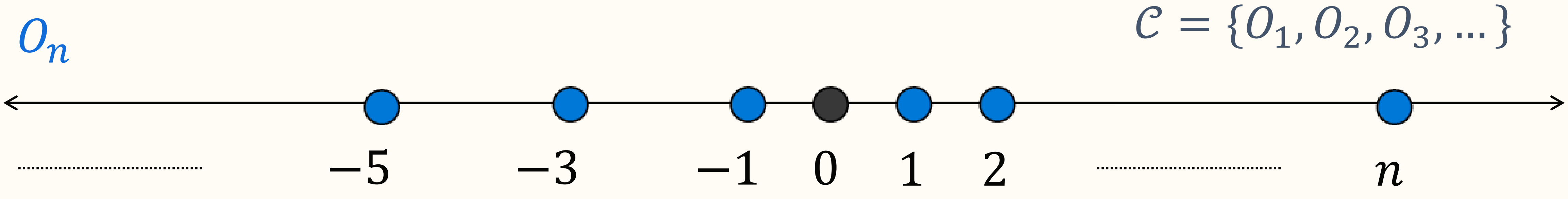
$\mathcal{C} = \{O_1, O_2, O_3, \dots\}$



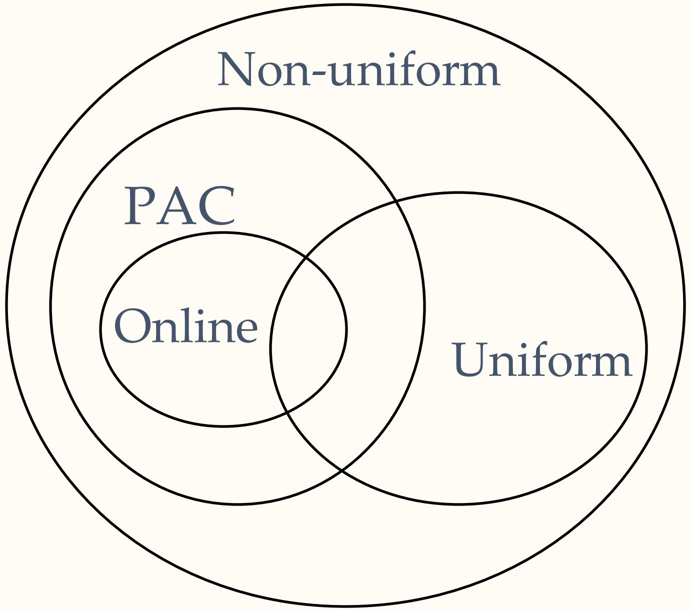
- Every language contains 0
- Uniformly generatable: $t^\star(\mathcal{C}) = 1$



Generation versus Prediction

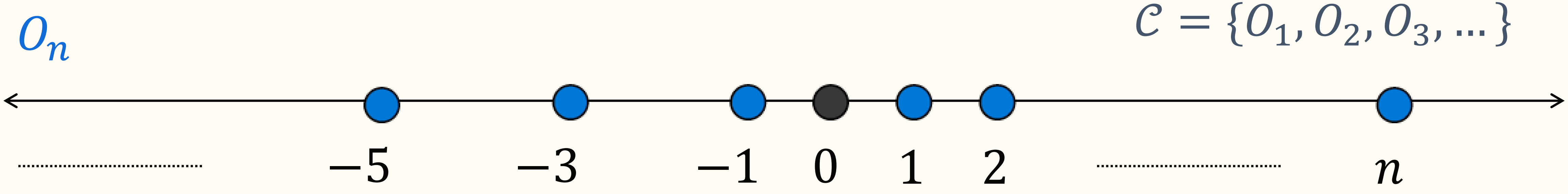


- Every language contains O
- Uniformly generatable: $t^{\star}(\mathcal{C}) = 1$
- Not too hard to see that VC dimension is some small constant $\Longrightarrow$ PAC learnable

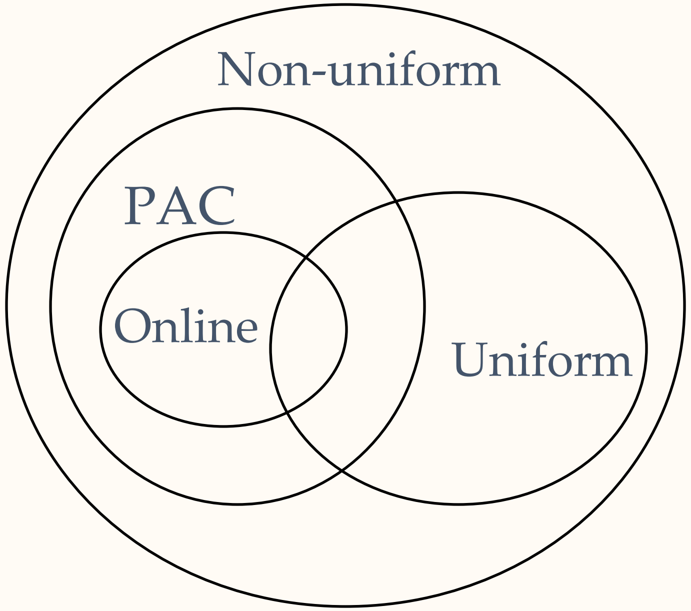


Generation versus Prediction

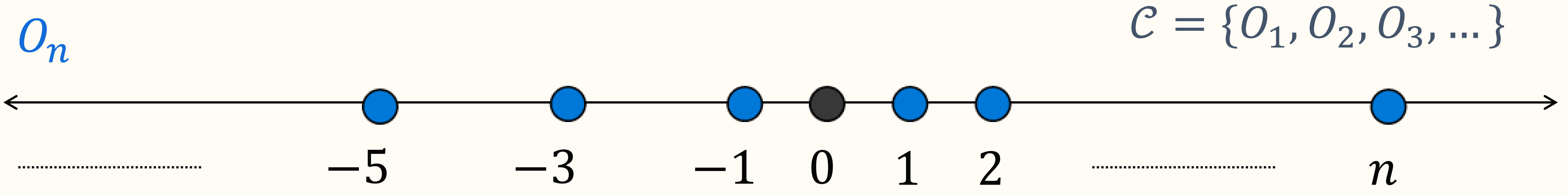
$\mathcal{C} = \{O_1, O_2, O_3, \dots\}$



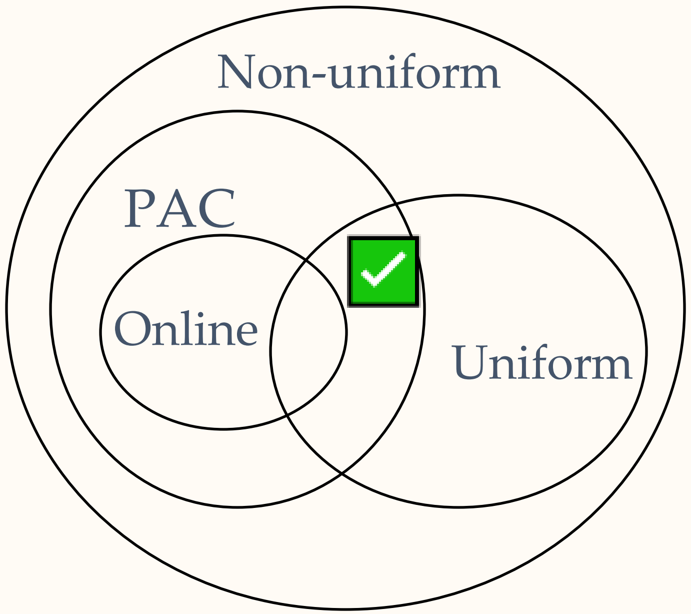
- Every language contains 0
- Uniformly generatable: $t^\star(\mathcal{C}) = 1$
- Not too hard to see that VC dimension is some small constant $\Longrightarrow$ PAC learnable
- Restriction on positive integers corresponds to thresholds $\Longrightarrow$ Not online learnable



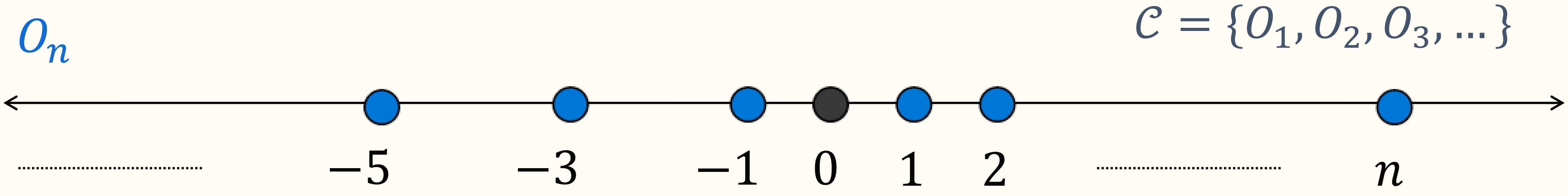
Generation versus Prediction



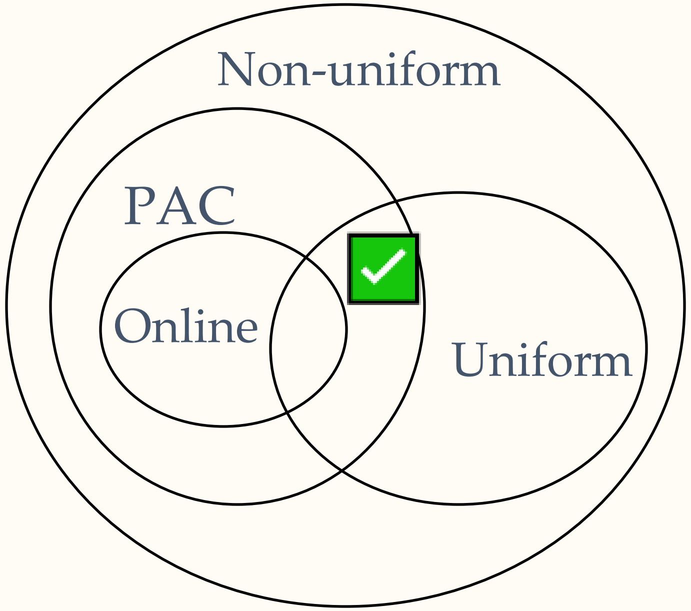
- Every language contains 0
- Uniformly generatable: $t^\star(\mathcal{C}) = 1$
- Not too hard to see that VC dimension is some small constant $\Longrightarrow$ PAC learnable
- Restriction on positive integers corresponds to thresholds $\Longrightarrow$ Not online learnable



Generation versus Prediction



- Every language contains O
- Uniformly generatable: $t^\star(\mathcal{C}) = 1$
- Not too hard to see that VC dimension is some small constant $\Longrightarrow$ PAC learnable
- Restriction on positive integers corresponds to thresholds $\Longrightarrow$ Not online learnable



recent work by Hanneke, Karbasi, Mehrotra, Velegkas '25 shows that generatability not even closed under finite unions! 😓

Statistical Model

Statistical Model

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$

Statistical Model

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, but instead of enumerating it in a *worst-case order*, chooses a distribution $\mathcal{D}$ supported on L_z

Statistical Model

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, but instead of enumerating it in a *worst-case order*, chooses a distribution $\mathcal{D}$ supported on L_z
- Generator $\mathcal{G}$ is shown a sequence of i.i.d samples from $\mathcal{D}$, and wishes to control

Statistical Model

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, but instead of enumerating it in a *worst-case order*, chooses a distribution $\mathcal{D}$ supported on L_z
- Generator $\mathcal{G}$ is shown a sequence of i.i.d samples from $\mathcal{D}$, and wishes to control

$$\Pr_{x_1,\dots,x_t \sim \mathcal{D}}[\mathcal{G}(x_1, \dots, x_t) \notin L_z \setminus \{x_1, \dots, x_t\}]$$

Statistical Model

- $\mathcal{C} = \{L_1, L_2, L_3, \ldots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, but instead of enumerating it in a *worst-case order*, chooses a distribution $\mathcal{D}$ supported on L_z
- Generator $\mathcal{G}$ is shown a sequence of i.i.d samples from $\mathcal{D}$, and wishes to control

valid, and not a repeat

$$\Pr_{x_1,\ldots,x_t \sim \mathcal{D}}[\mathcal{G}(x_1, \ldots, x_t) \notin L_z \setminus \{x_1, \ldots, x_t\}]$$

Statistical Model

- $\mathcal{C} = \{L_1, L_2, L_3, \dots\}$, each L_i countably infinite subset of universe $\mathcal{X}$
- Adversary chooses some target language L_z, but instead of enumerating it in a *worst-case order*, chooses a distribution $\mathcal{D}$ supported on L_z
- Generator $\mathcal{G}$ is shown a sequence of i.i.d samples from $\mathcal{D}$, and wishes to control

valid, and not a repeat

$$\Pr_{x_1,\dots,x_t \sim \mathcal{D}}[\mathcal{G}(x_1, \dots, x_t) \notin L_z \setminus \{x_1, \dots, x_t\}]$$

- Kalavasis, Mehrotra, Velegkas '24 show that for the Kleinberg, Mullainathan '24 algorithm, this is bounded as $C \cdot \exp(-c \cdot t)$ for all countable collections!

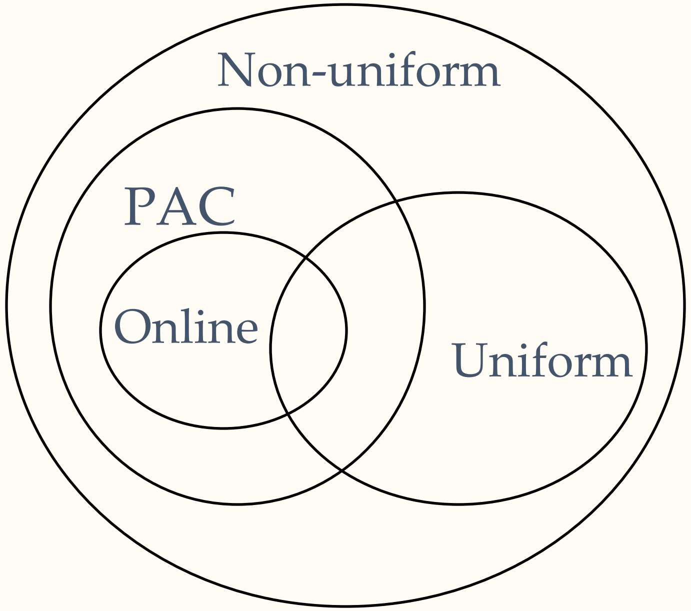
Summary

Summary

- Non-uniform / uniform generation in the limit : stronger requirements than vanilla generation

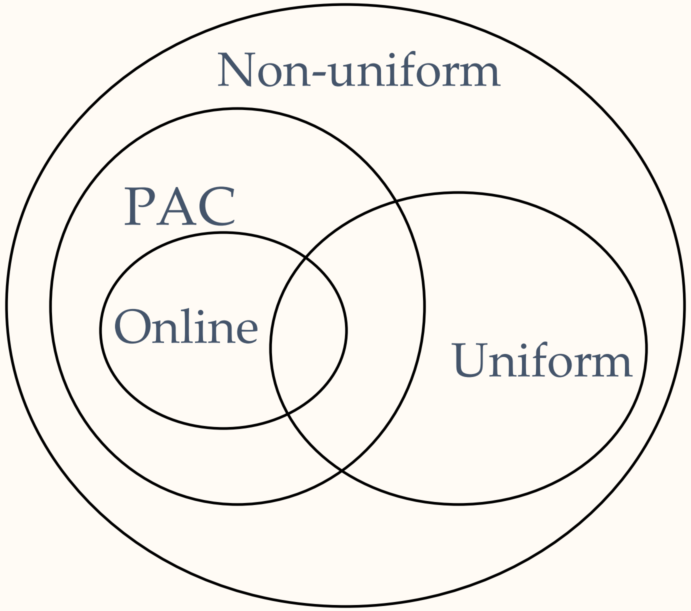
Summary

- Non-uniform / uniform generation in the limit : stronger requirements than vanilla generation
- Generation incomparable to prediction



Summary

- Non-uniform / uniform generation in the limit : stronger requirements than vanilla generation
- Generation incomparable to prediction
- <u>Open:</u> complete characterization for generation in the limit



Summary

- Non-uniform / uniform generation in the limit : stronger requirements than vanilla generation
- Generation incomparable to prediction
- <u>Open:</u> complete characterization for generation in the limit
- <u>Open:</u> black-box transforming algorithm that generates in the limit $\Longrightarrow$ exponential rate