Energy Complexity of Fully-Connected Layers

Jiří Šíma

sima@cs.cas.cz

Institute of Computer Science
Czech Academy of Sciences, Prague, Czechia

joint work with **Jérémie Cabessa**

jeremie.cabessa@uvsq.fr

DAVID Laboratory
Paris-Saclay University, Versailles, France

Efficient Processing of Deep Neural Networks (DNNs)

- DNNs are widely used for many artificial intelligence (AI) applications including computer vision, speech recognition, natural language processing, robotics etc.
- DNNs achieve state-of-the-art accuracy on many AI tasks at the cost of high computational complexity (tens of millions of operations for a single inference)
- energy efficiency of DNN implementations in low-power hardware operated on batteries (e.g. cellphones, smartwatches, smart glasses) becomes crucial

$\longrightarrow$ reducing the energy cost of DNNs:

1. approximate computing methods (e.g. low floating-point precision, approximate multipliers) in error-tolerant applications such as image classification
2. hardware design: energy-efficient implementations of DNNs on various hardware platforms including GPUs, FPGAs, in-memory computing architectures

Energy Consumption of DNNs

- the power consumption of a specific DNN hardware implementation can be measured or calculated/estimated (using physical laws)
- a plethora of methods that minimize the energy consumption of a given DNN on various hardware architectures
(Sze,Chen,Yang,Emer:Efficient Processing of Deep Neural Networks,2020)
- automated by software tools, for example, the `Timeloop` program maps a convolutional layer specified by its parameters onto a given hardware architecture (e.g. Simba, Eyeriss) that is optimal in terms of power consumption estimated by `Accelergy` tool which reports the energy statistics
- it has been empirically observed that the energy for DNN inference is mainly consumed by
  1. data movement inside a memory hierarchy (approx. 70%) corresponding to the data energy E_{data}
  2. multiply-and-accumulate (MAC) operations (approx. 30%): $S \leftarrow S + wx$ on floats S, w, x, corresponding to the computation energy E_{comp}

$$\longrightarrow \quad E = E_{\text{data}} + E_{\text{comp}}$$

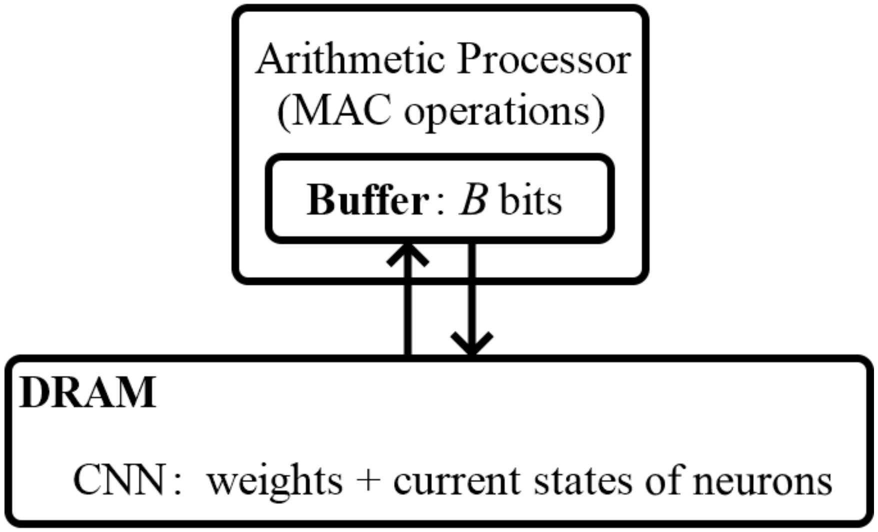
Motivations for Energy Complexity Model of DNNs

- formal computational models are fundamental for defining robust complexity measures and classes, e.g. Turing machine for efficient (polynomial-time) computations characterized by the complexity class P (vs. NP)
- energy as a new computational resource alternative to computation time and memory space which are quantified asymptotically using Big O notation
- lower bounds on computational complexity establish principal limits of efficient algorithms

$\longrightarrow$ Simplified Hardware-Independent Model of Energy Complexity for DNNs:

- abstracts from hardware implementation details, ignoring specific aspects and parameters of real-world machine
- preserves the asymptotic energy of DNN inference
- focuses, for simplicity, on separate convolutional layers, avoiding global energy optimization across multiple CNN (convolutional neural network) layers

Energy Complexity Model for CNNs (Šíma,Vidnerová,Mrázek,2023)



- only two memory levels called DRAM (large, slow, and cheap memory) and Buffer of limited capacity B bits (small, fast, and expensive memory)
- CNN weights and states are stored in DRAM
- arithmetic operations are performed over numerical data stored in Buffer
- the dataflow controls the transfer of data between DRAM and Buffer
- the main idea: the three arguments stored in DRAM, input x, weight w, and accumulated output S of each MAC operation $S \leftarrow S + wx$ performed for evaluating a given convolutional layer, must occur in Buffer simultaneously

Energy Complexity Measure $E = E_{\text{data}} + E_{\text{comp}}$

for a given dataflow:

E_{data} is proportional to the number of DRAM accesses
E_{comp} is proportional to the number of MACs over data in `Buffer`

Example: the dataflow with write-once outputs: each output of a single neuron is completely evaluated at once in `Buffer` before writing to `DRAM`

its theoretical energy complexity E_{data} in terms of convolutional layer parameters:

$E_{\text{data}} = O(d)$ where d is the layer depth (the number of feature maps)

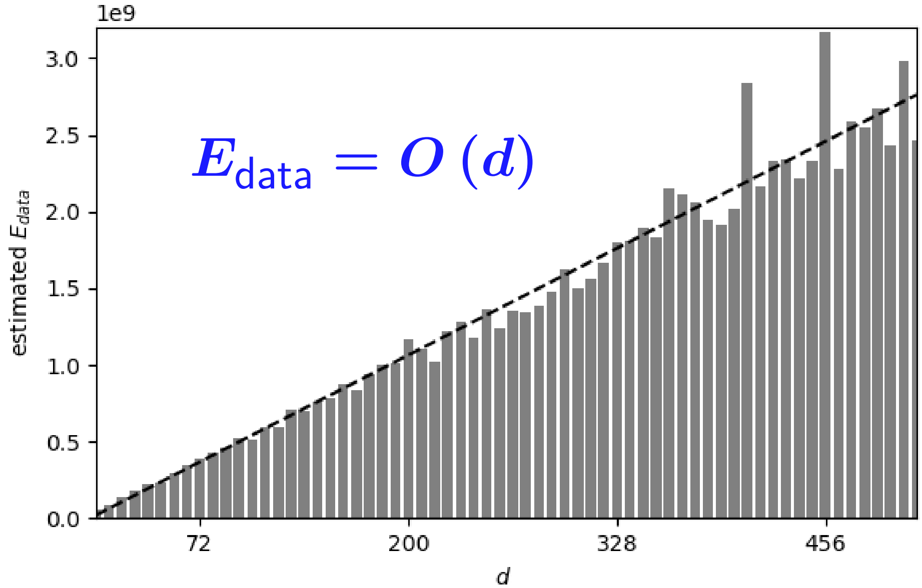
$E_{\text{data}} = O(h^2)$ where h is the layer height=width (the size of feature maps)

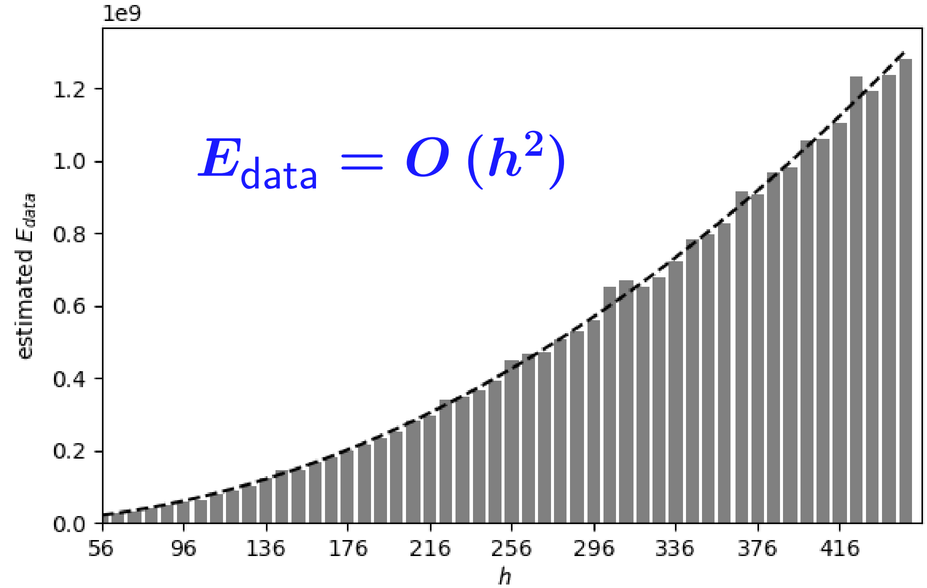
$E_{\text{data}} = O(r^2)$ where r is the size of receptive fields

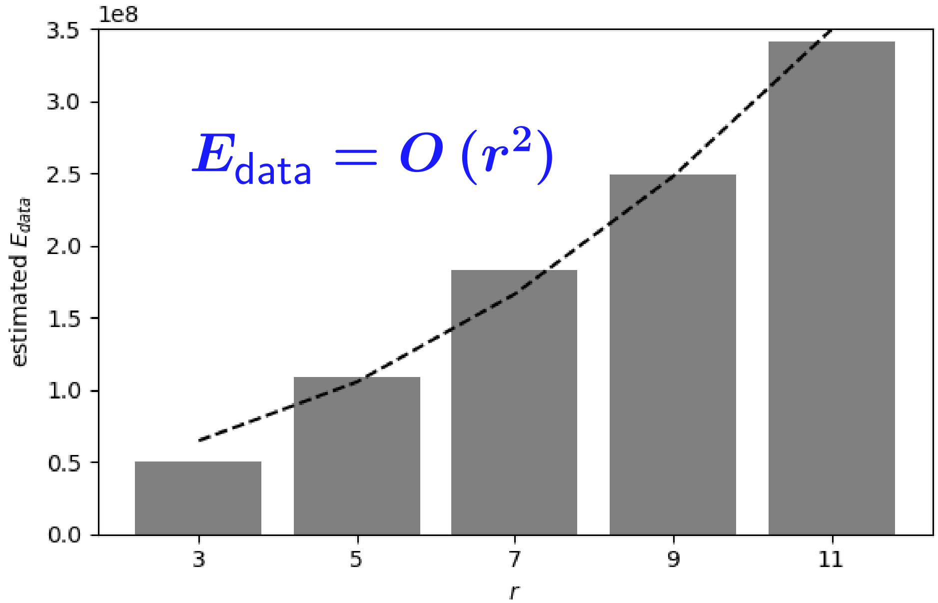
$E_{\text{data}} = O(\sigma^{-2})$ where σ is the stride

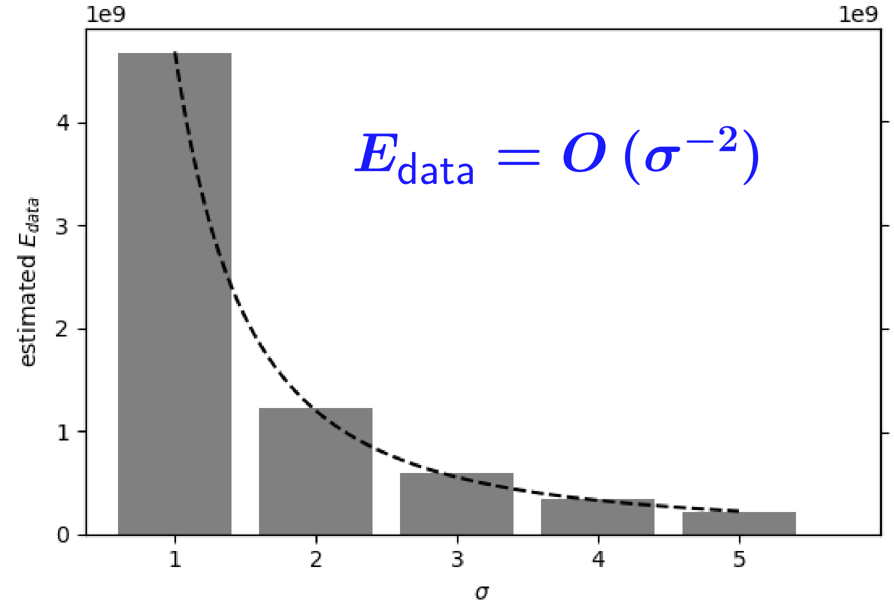
fits very well (by linearity/quadraticity statistical tests) the real power consumptions estimated by the `Timeloop/Accelergy` software platform that maps a convolutional layer of given parameters onto the Simba and Eyeriss hardware architectures:

Experimental Validation of Energy Complexity Model for Simba

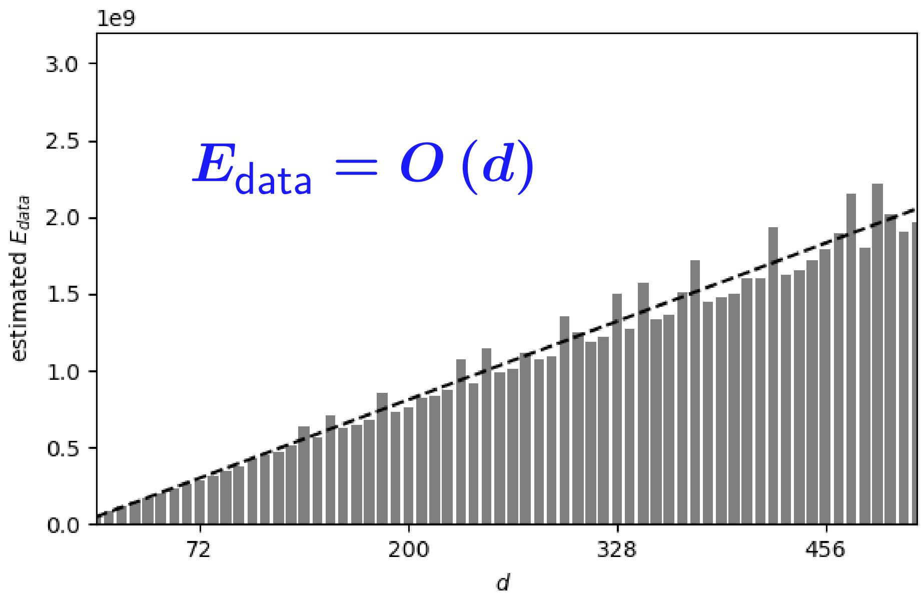


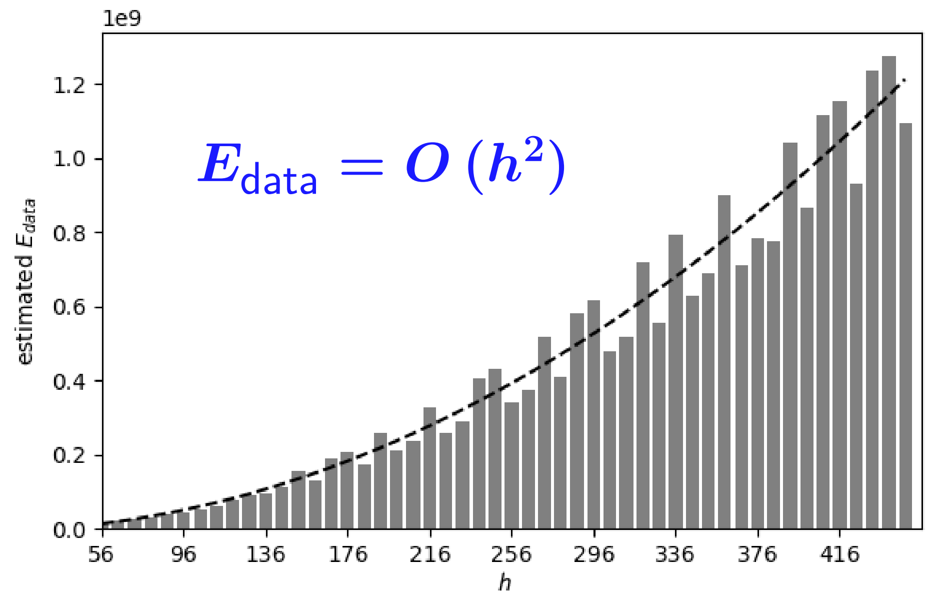


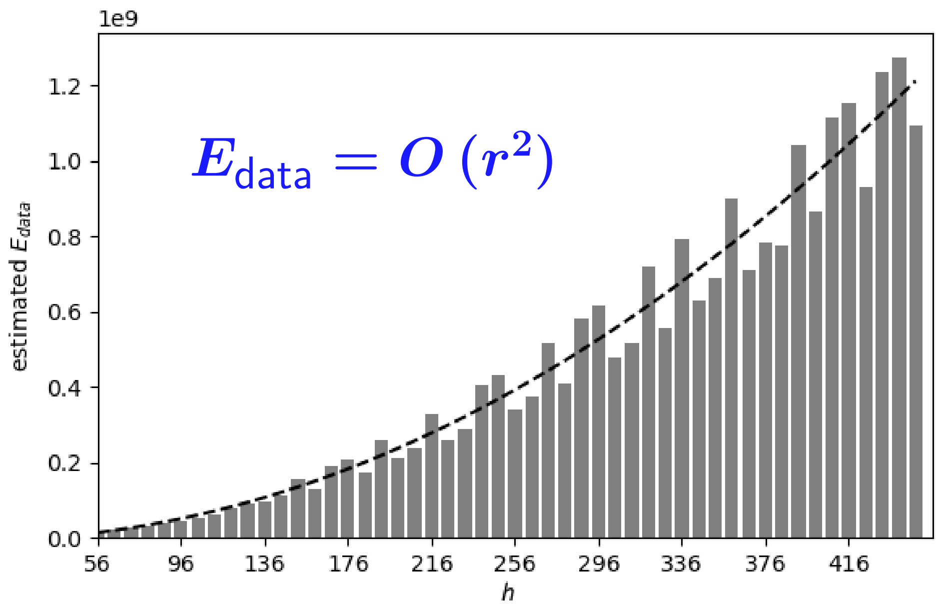


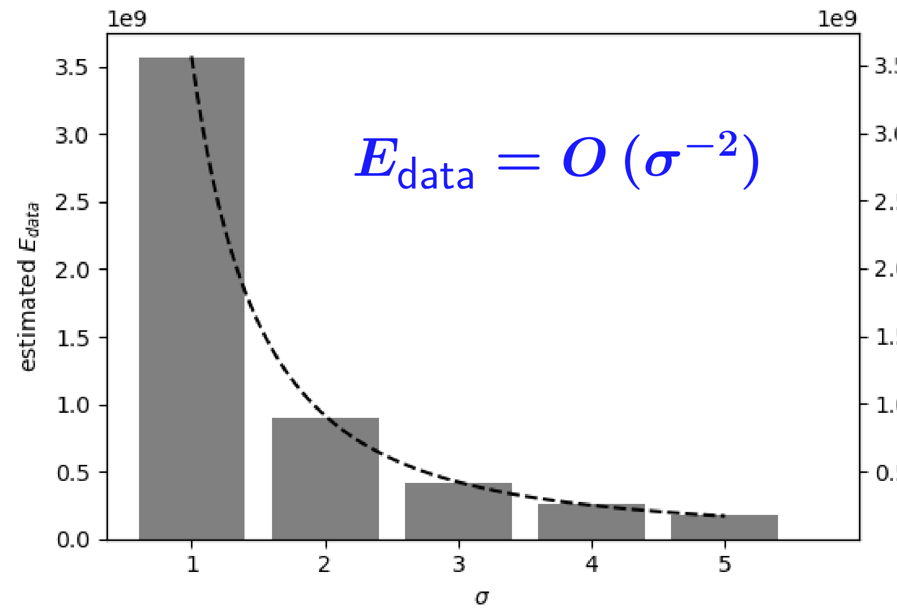


Experimental Validation of Energy Complexity Model for Eyeriss

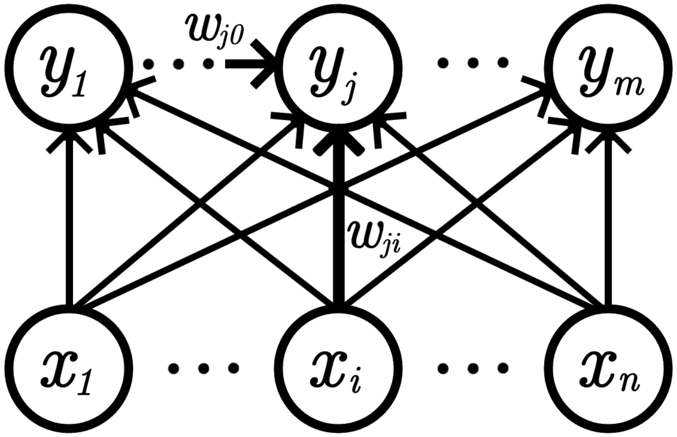








Energy Complexity of Fully-Connected (FC) Layers



$$y_j = \text{ReLU}\left(w_{j0} + \sum_{i=1}^{n} w_{ji} x_i\right)$$

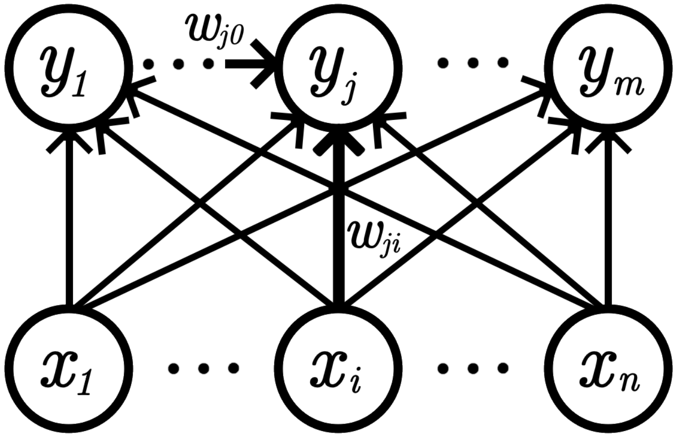
for every $j = 1, \ldots, m$

1. Computation Energy: each of the m outputs is initialized with bias w_{j0} and requires n MAC updates

$$\longrightarrow \quad E_{\text{comp}} = C_b\, mn$$

where C_b is a non-uniform constant specific to b-bit MAC circuit inside a microprocessor

Energy Complexity of Fully-Connected (FC) Layers



$$y_j = \text{ReLU}\left(w_{j0} + \sum_{i=1}^{n} w_{ji} x_i\right)$$

for every $j = 1, \ldots, m$

2. Data Energy: we count DRAM accesses for weights, outputs, and inputs separately $\longrightarrow$ $E_{\text{data}} = E_{\text{weights}} + E_{\text{outputs}} + E_{\text{inputs}}$

- for each of the mn pairs of inputs x_i and (accumulated) outputs y_j (partial sums) that occurs in `Buffer`, the corresponding unique weight w_{ji} is read once
- each output read into `Buffer` is later written to `DRAM`

$\longrightarrow$ $E_{\text{data}} = b\,(mn + 2\mu + \nu)$ (it thus suffices to minimize $2\mu + \nu$)

where b is the number of bits in the float representation;

μ and ν is the number of DRAM accesses to read outputs and inputs, respectively

A Simple General Lower Bound on Data Energy Complexity

assumption: the `Buffer` capacity is $B = b(\beta + 1)$ bits

where $\beta > 1$ floating-point numbers of size b bits are used for inputs and outputs while the remaining one serves for weights

observation: we get at most $\beta - 1$ input-output pairs by reading one input/output into `Buffer` $\times$ all the mn pairs need to meet in `Buffer`

$\longrightarrow$ $\mu + \nu \geq \frac{mn}{\beta - 1}$ DRAM reads & we know $\mu \geq m$

the trivial lower bound on the data energy follows:

$$E_{\text{data}} = b\,(mn + 2\mu + \nu) \geq b\left(mn + \frac{mn}{\beta - 1} + m\right)$$

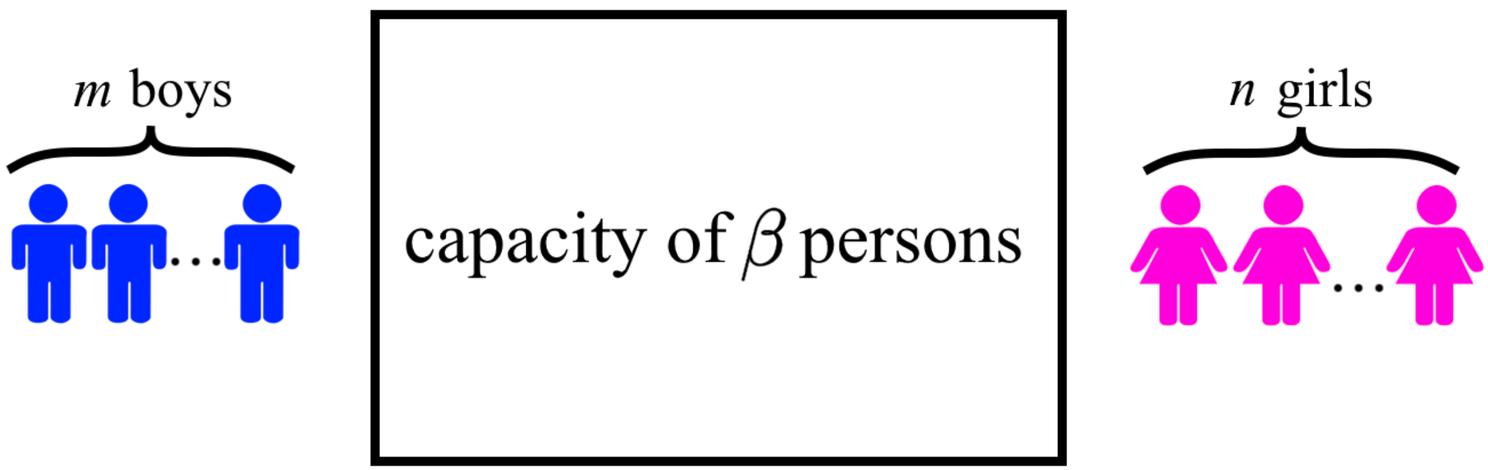
which can slightly be improved in general case:

$$E_{\text{data}} \geq b\left(mn + \frac{mn}{\beta - 1} + m + \frac{\beta - 2}{(\beta - 1)^2}\min(m, n) + 1\right)$$

Meeting of All Pairs in a Limited-Capacity Room

popular formulation of the data energy problem for FC layers
(m outputs $\equiv$ boys, n inputs $\equiv$ girls, `Buffer` $\equiv$ room of capacity β persons, $\mu + \nu$ DRAM reads $\equiv$ boy $+$ girl entrances):

What is the smallest number $\mu + \nu$ of person entrances in a room that can hold at most β people, so that each of the m boys meets each of the n girls in that room **?** (only one person can enter the room at a time, replacing someone inside if the room is full)

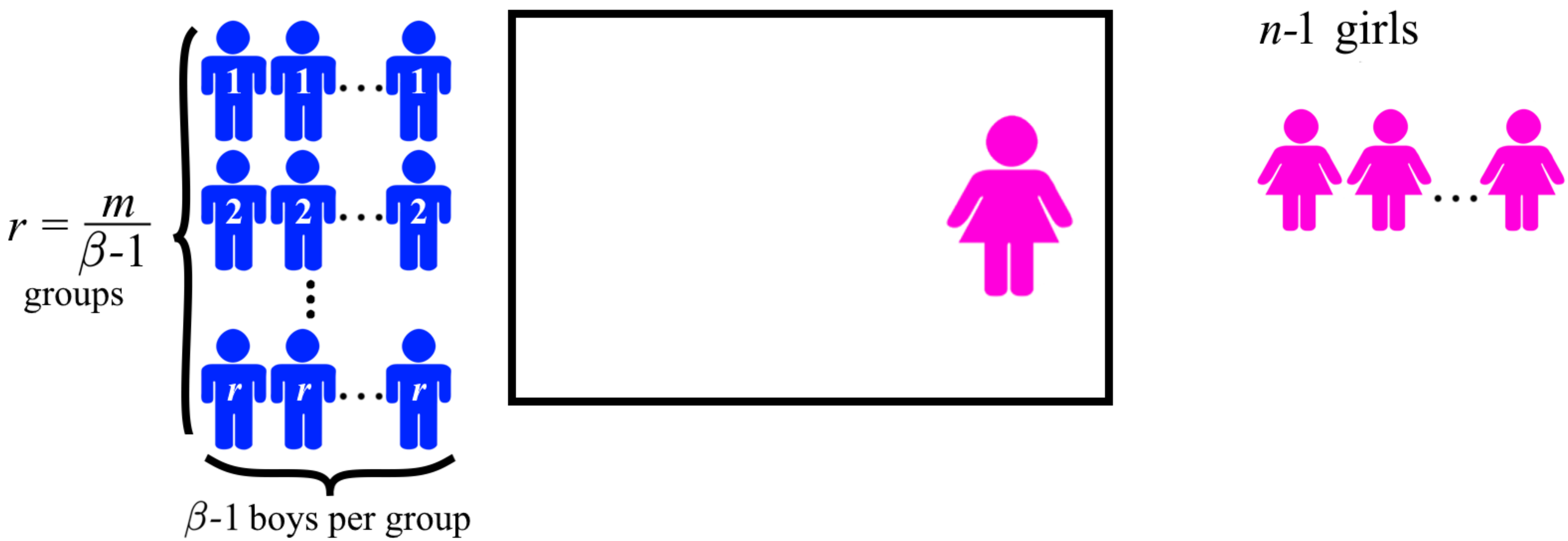


An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room

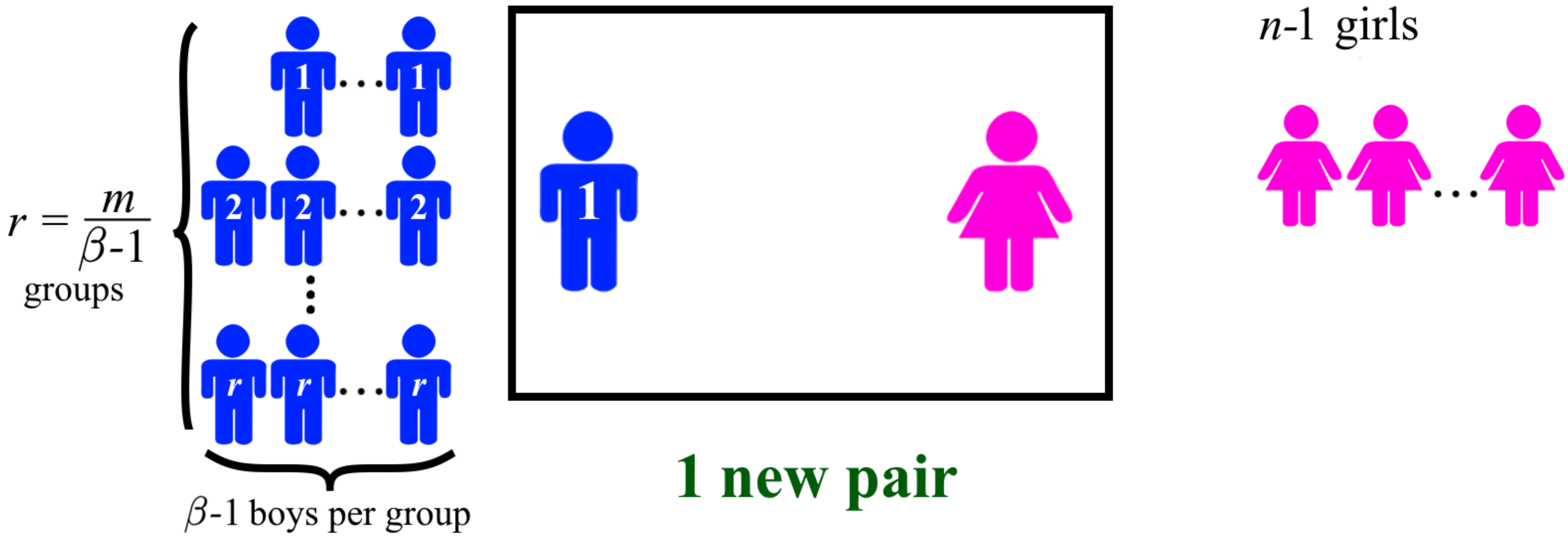
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



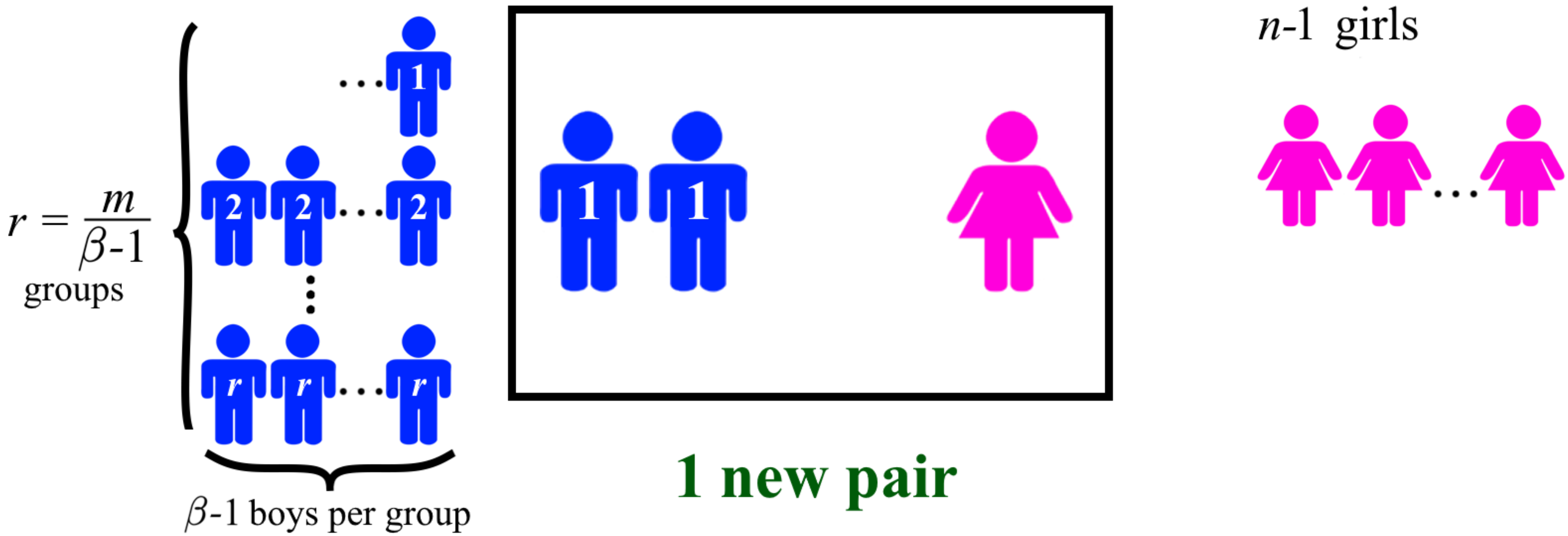
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



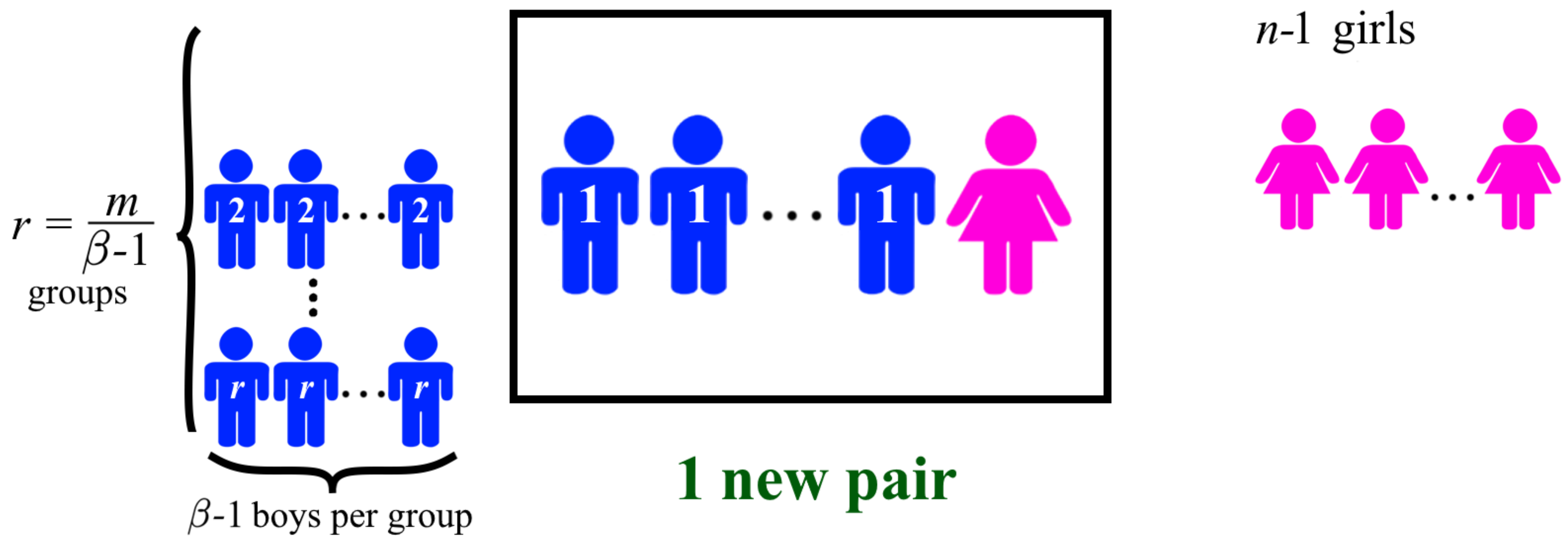
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



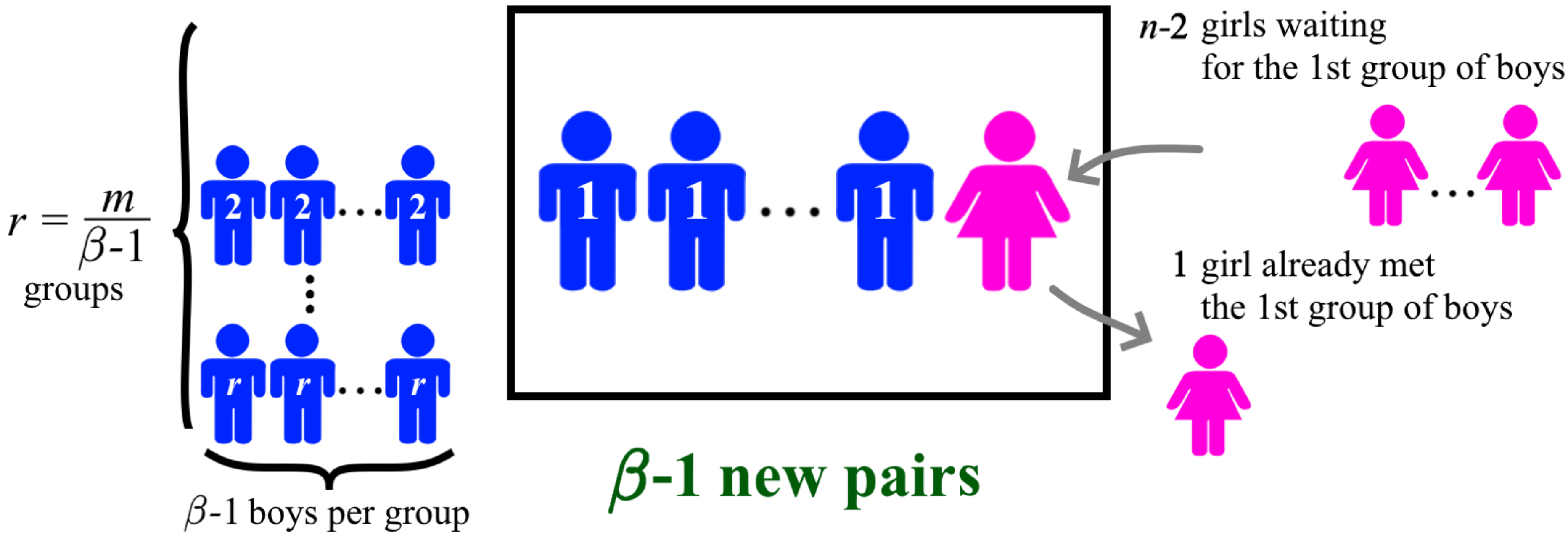
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



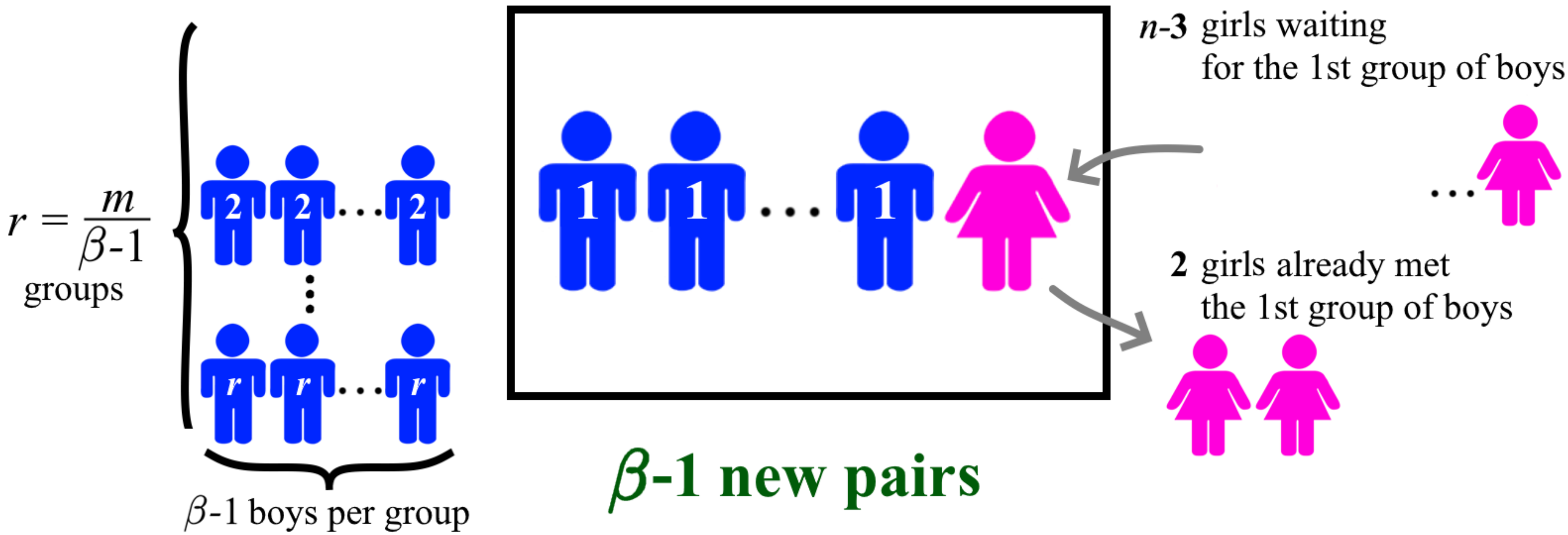
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



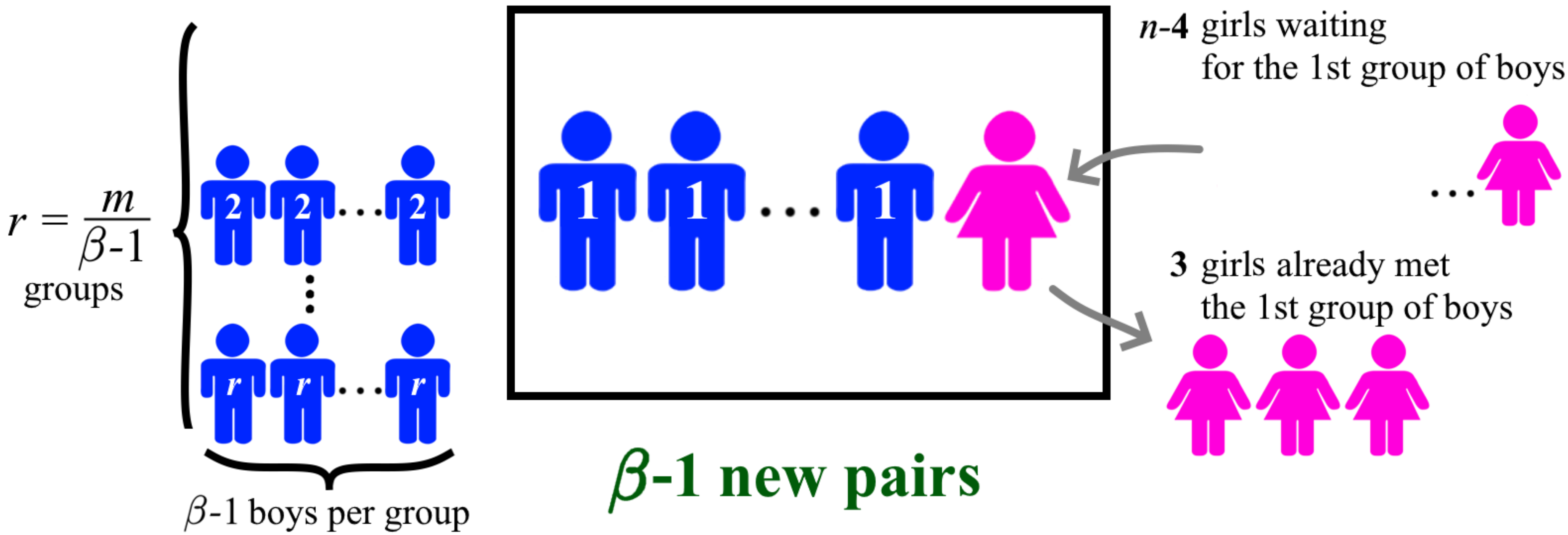
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



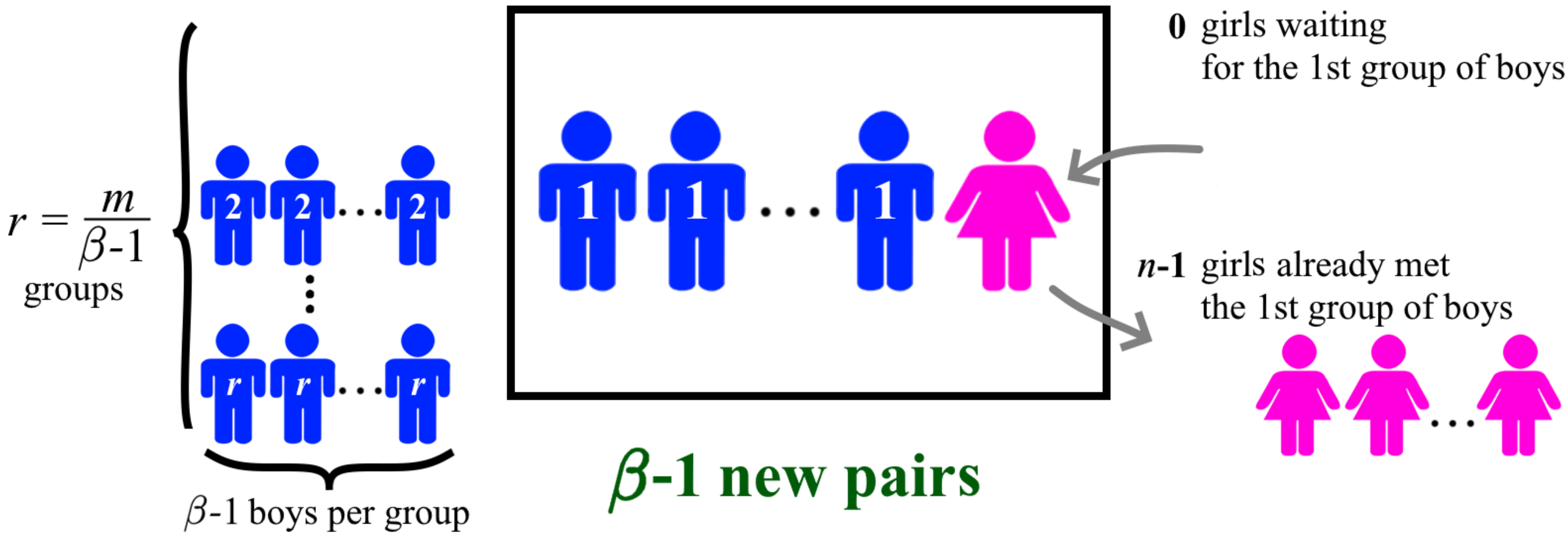
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



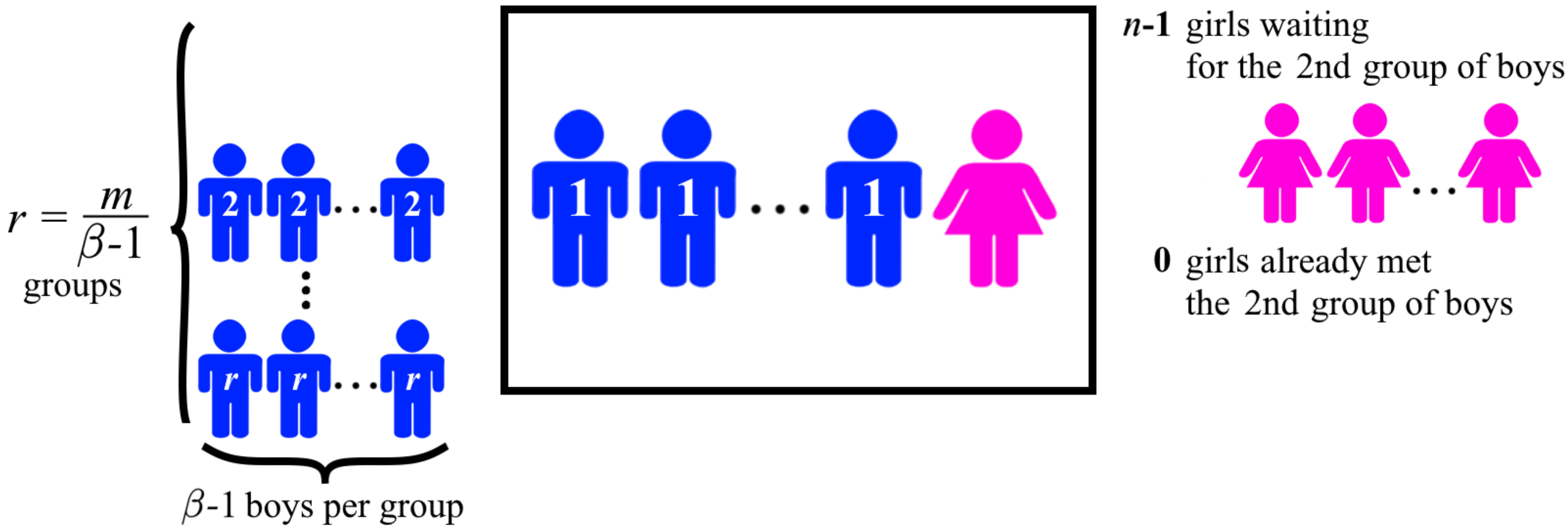
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



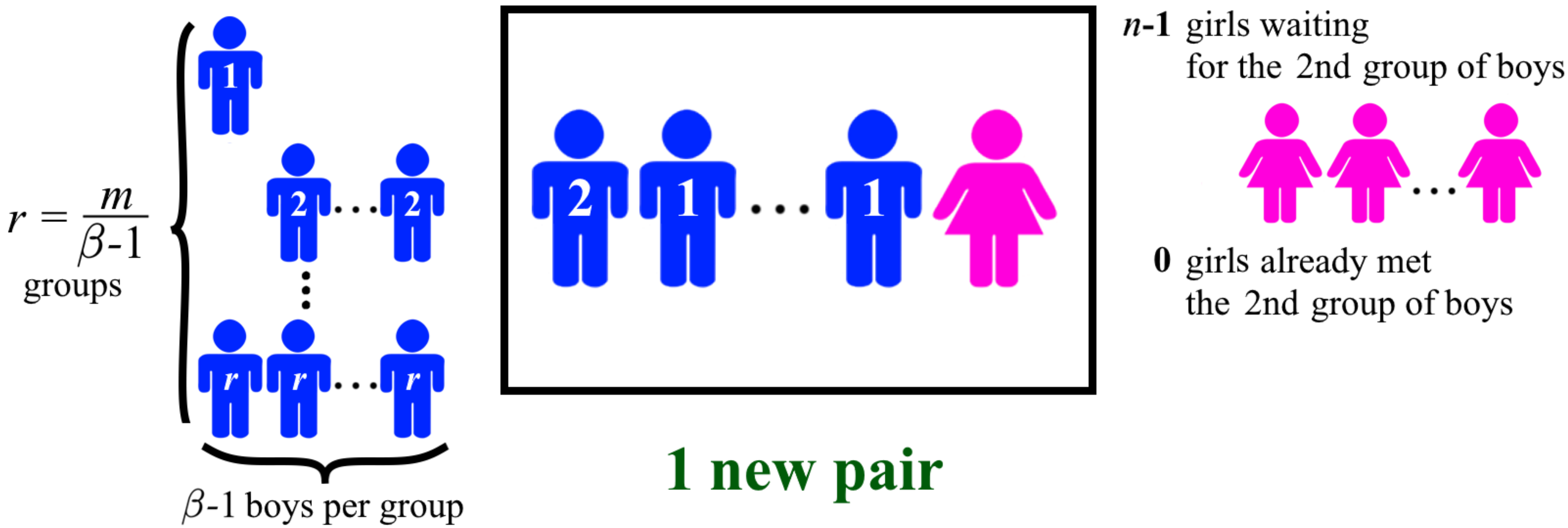
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



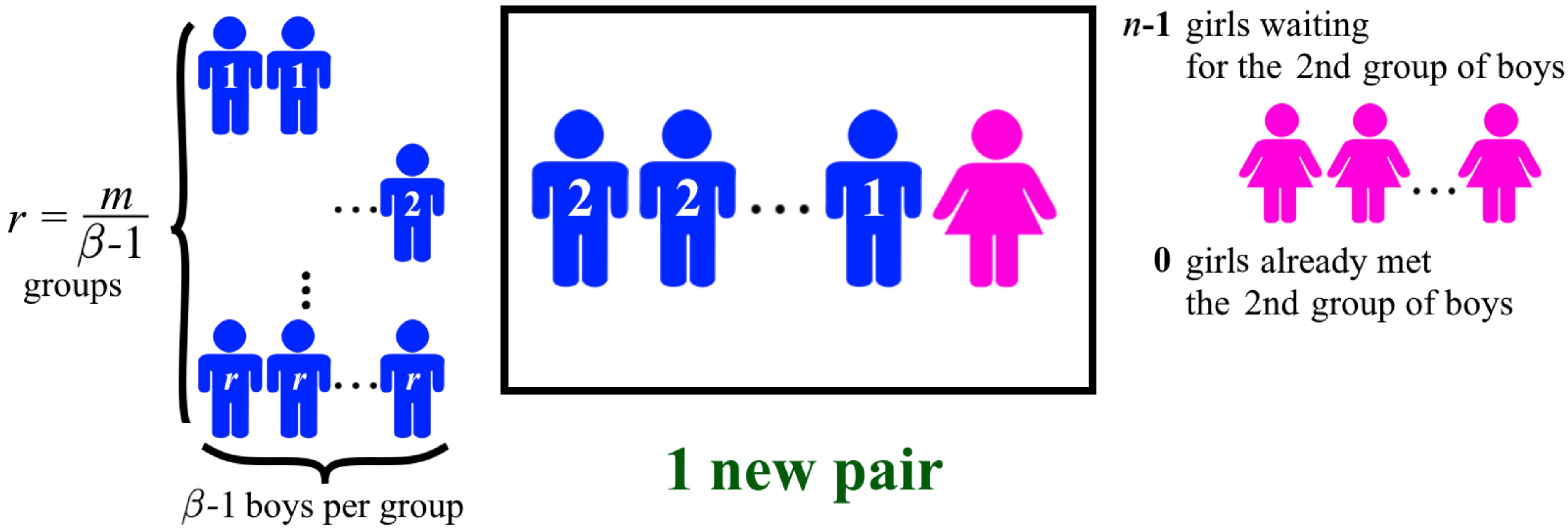
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



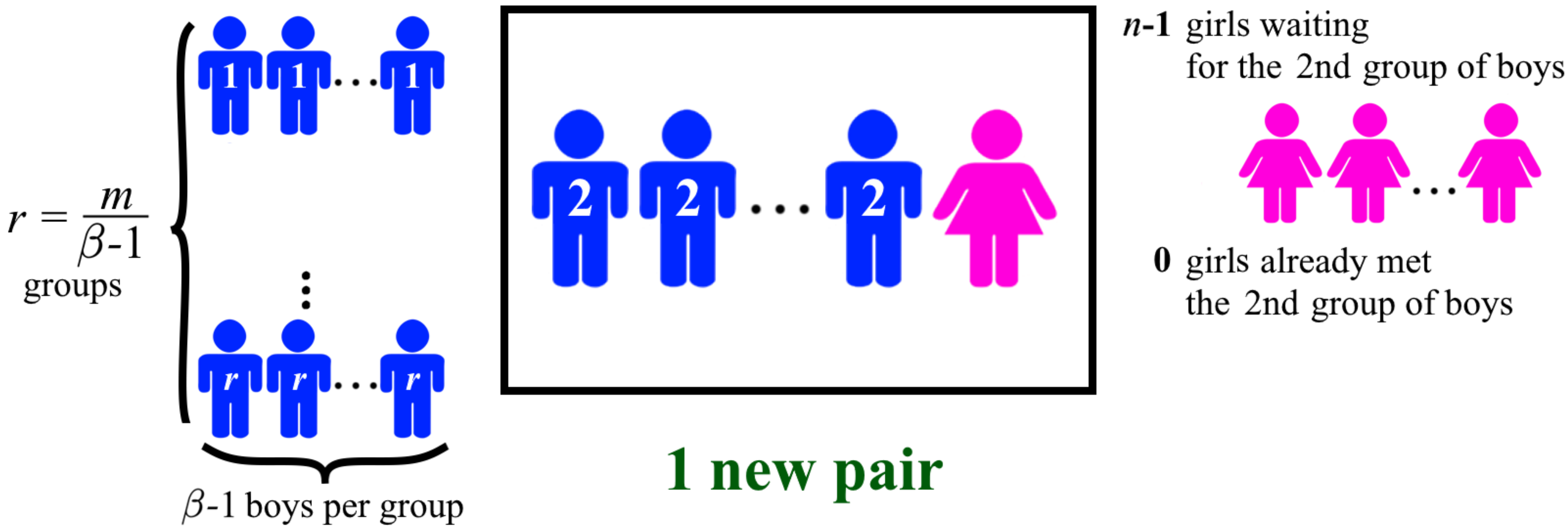
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



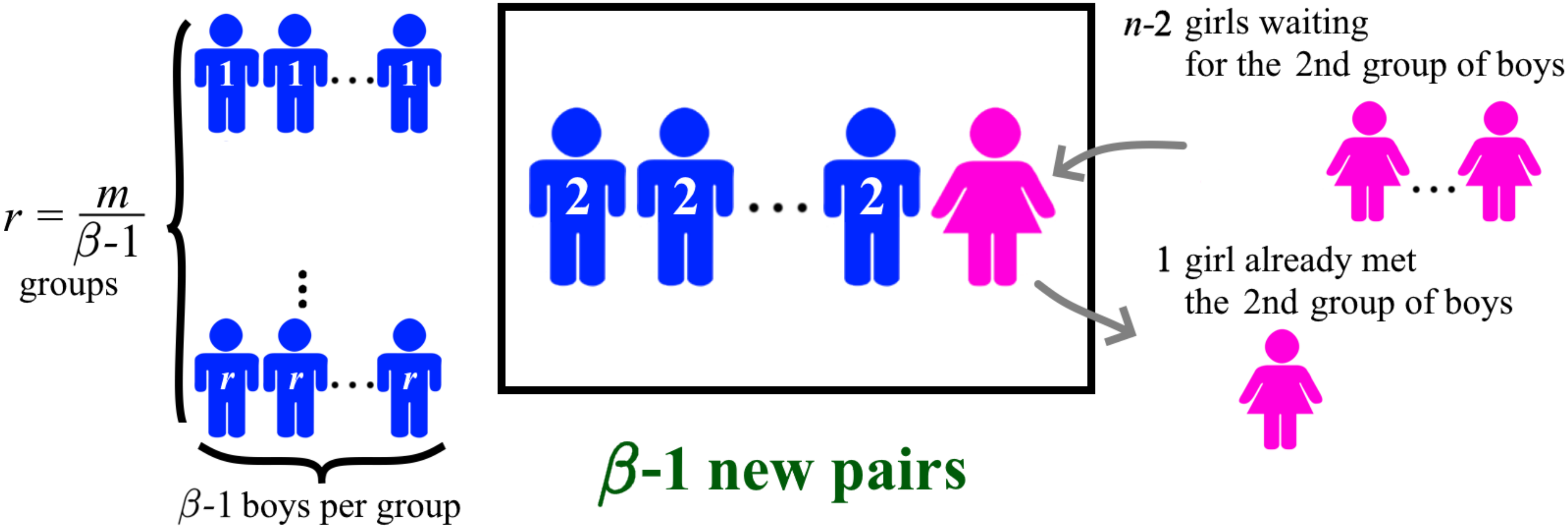
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



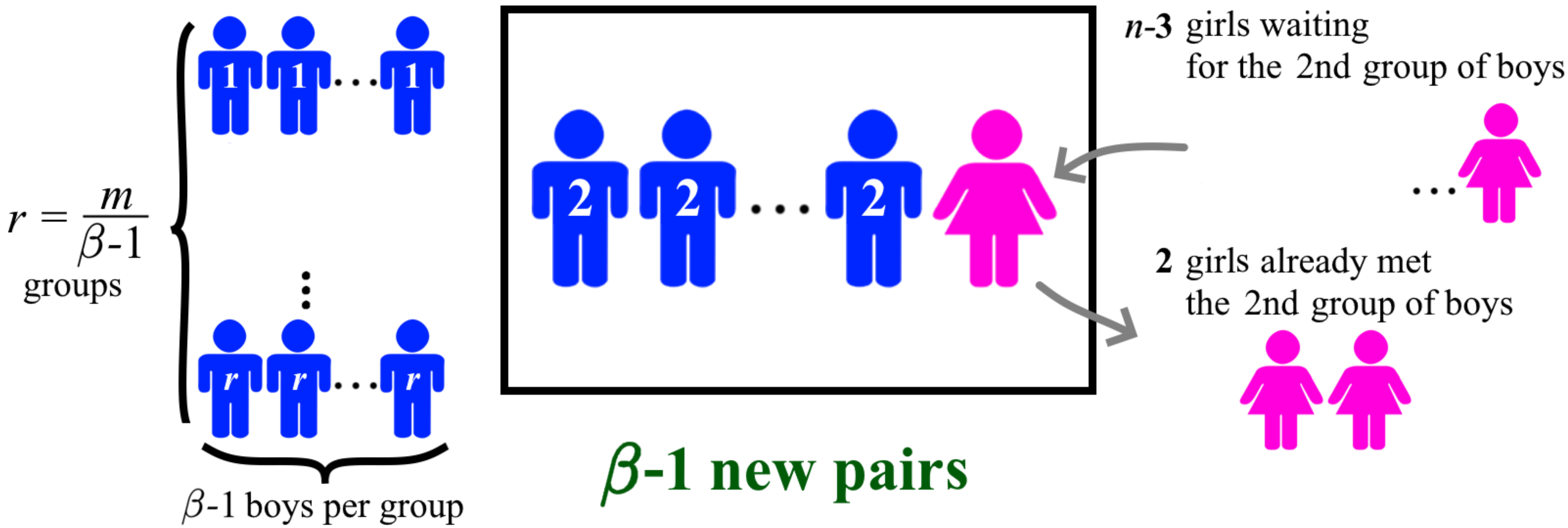
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



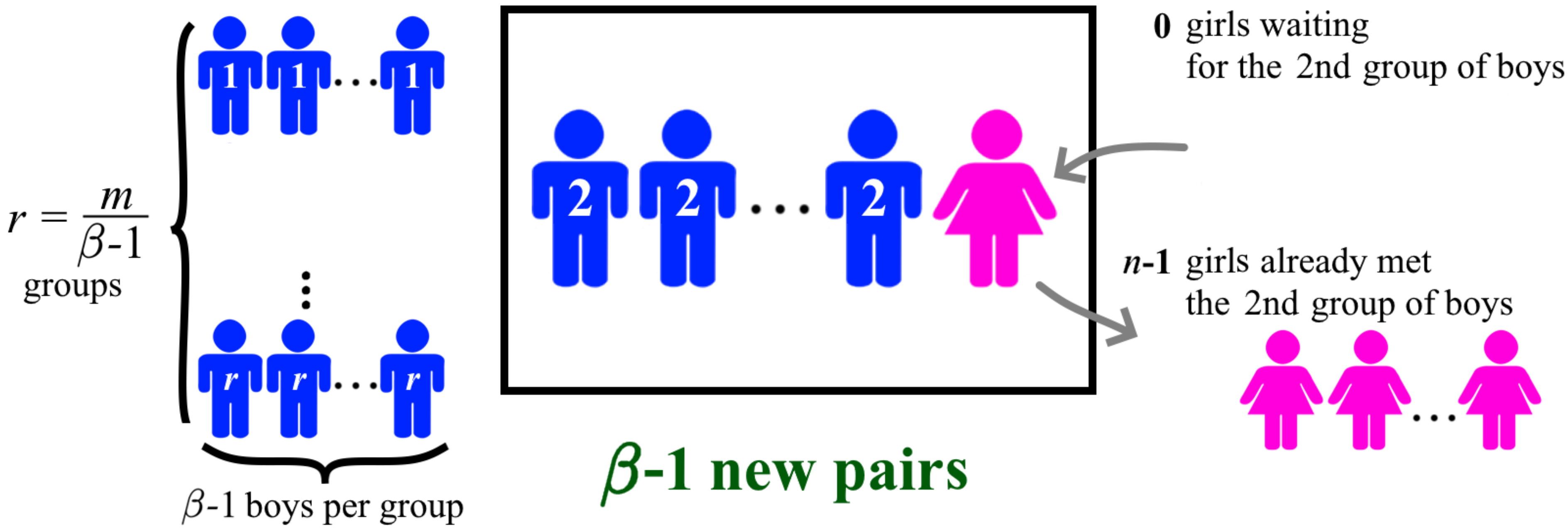
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



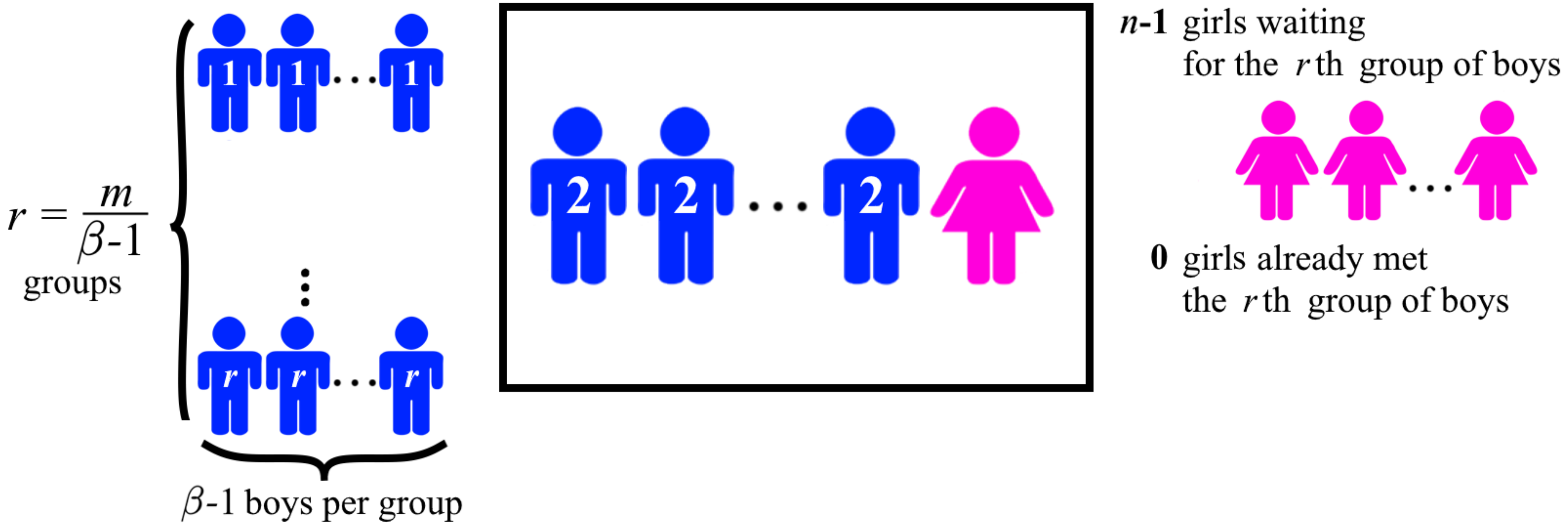
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



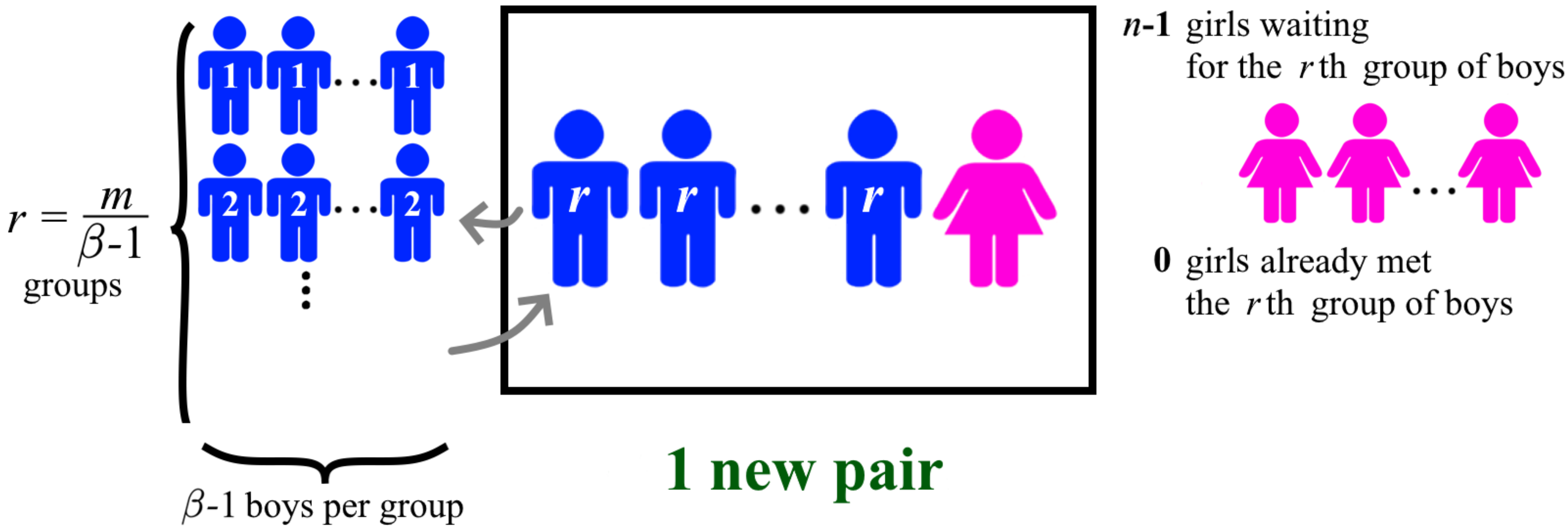
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



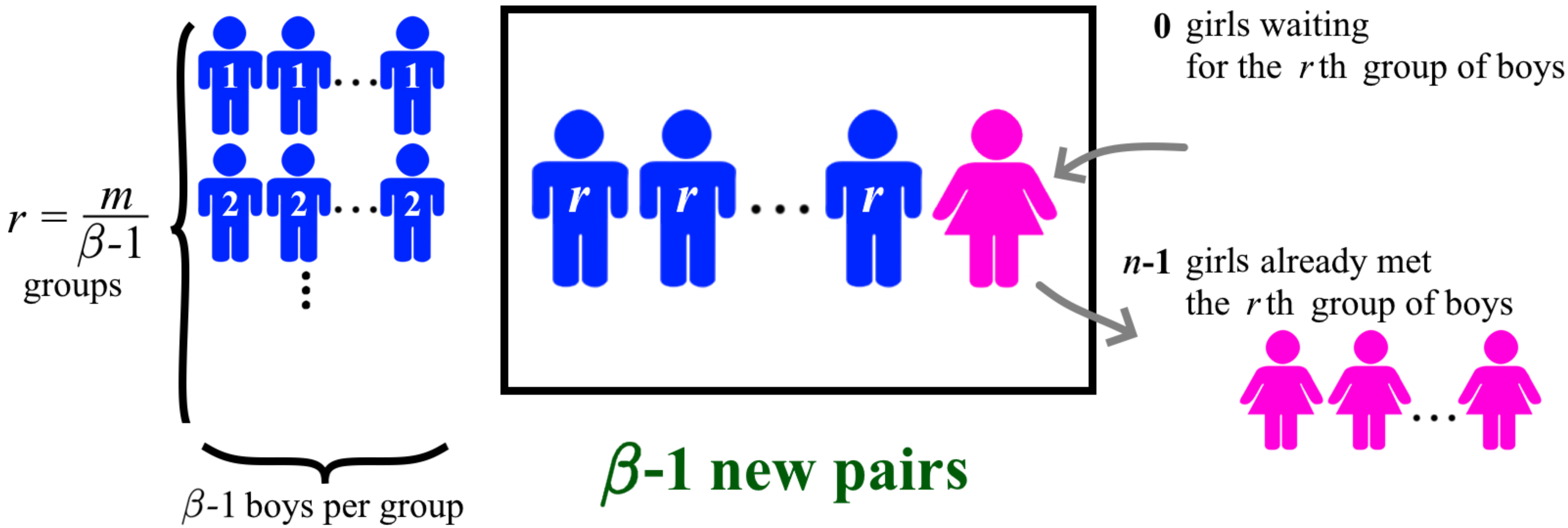
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



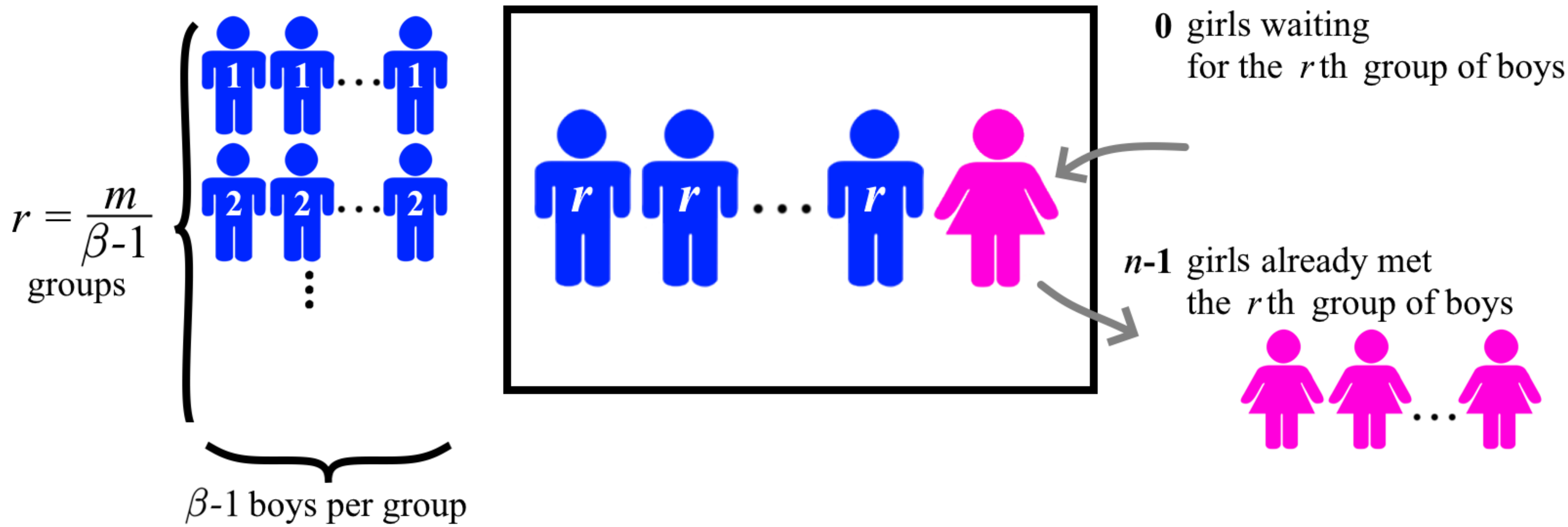
An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room



$\longrightarrow$ $\boldsymbol{\mu = m}$ boy entrances & $\boldsymbol{\nu = \frac{m}{\beta - 1}\Big(n - 1\Big) + 1}$ girl entrances

An Upper Bound on Data Energy Complexity of FC layers

the dataflow by solving the problem of meeting pairs in a limited-capacity room

$$E_{\text{data}} = b\,(mn + 2\mu + \nu)$$

where $$2\mu + \nu = \frac{m}{\beta - 1}\Big(n - 1\Big) + 2m + 1$$

$$\longrightarrow \quad E_{\text{data}} \leq b\left(mn + \frac{m(n-1)}{\beta - 1} + 2m + 1\right)$$

cf. $$E_{\text{data}} \geq b\left(mn + \frac{m(n-1)}{\beta - 1} + 2m + 1 - \frac{\beta - 2}{\beta - 1}\left(m - \frac{\min(m,n)}{\beta - 1}\right)\right)$$

Optimal Energy Complexity for Partitioned `Buffer`

`Buffer` is divided into two fixed parts separated for d inputs and $\beta - d$ outputs

Example: $d = 1$ (similarly for arbitrary $1 \leq d \leq \beta - 1$)

1. Linear Program formulation: find $\mu \geq 0$ and $\nu \geq 0$ that minimize $2\mu + \nu$

   subject to $\mu + (\beta - 1)\,\nu \geq mn$ (at most 1 or $\beta - 1$ new pairs by reading one output or input, respectively)

   and $\mu \geq m$ (at least m outputs are read)

2. Dual Linear Program: find $\phi \geq 0$ and $\psi \geq 0$ that maximize $mn\,\phi + m\,\psi$

   subject to $\phi + \psi \leq 2$ and $(\beta - 1)\,\phi \leq 1$

   which has a feasible solution $\phi_0 = \frac{1}{\beta - 1}$ and $\psi_0 = 2 - \frac{1}{\beta - 1}$

the matching lower bound by the weak duality theorem:

$$2\mu + \nu \geq mn\,\phi_0 + m\,\psi_0 = \frac{m(n-1)}{\beta - 1} + 2m$$

$\longrightarrow$ optimal energy complexity $E_{\text{data}} = b\left(mn + \dfrac{m(n-1)}{\beta - 1} + 2m + 1\right)$

(can also be proven in some other special cases of contiguous `Buffer`)

A Summary

- In our previous work, we have introduced a machine-independent model of energy complexity for CNNs, which fits very well the power consumption estimates of various CNN hardware implementations.
- As a starting point for convolutional layers, we have theoretically analyze the energy complexity model for FC layers proposing an energy-efficient dataflow which provides an upper bound on energy complexity of FC layers.
- We have proven the optimal energy complexity of FC layers for partitioned `Buffer`.

Open Problems

- the matching lower bound on energy of FC layers for contiguous `Buffer` **?**
- the optimal energy complexity for convolutional layers **?**