Reliability

- **State-of-the-art:** Bootstrap confidence intervals ("error bars") around evaluation scores under meta-parameter variation.

  [Lucic et al., 2018, Henderson et al., 2018]
- **Goal:**
  - Analyze sources of variability in performance evaluation,
  - analyze interaction of meta-parameters variance with data properties,
  - compute coefficient to quantify general robustness of a model.
- **Method:**
  - **Variance component analysis (VCA)**: Untangle sources of variability in measurement.
  - **Reliability coefficient**: Assess general robustness of model by ratio of substantial variance out of total variance.

VCA in classical ANOVA [Fisher, 1925, Searle et al., 1992]

- Example: Specify model with random effects for variation in outcome Y between sentences s and between settings of meta-parameter r.
- **Tautological decomposition:**

$$Y = \mu + (\mu_s - \mu) + (\mu_r - \mu) + (Y - \mu_s - \mu_r + \mu),$$

  - grand mean μ of observed evaluation score across all levels of meta-parameter r and sentences s,
  - deviation $\nu_s = (\mu_s - \mu)$ of mean score μ_s for sentence s from μ,
  - deviation $\nu_r = (\mu_r - \mu)$ of mean score μ_r for meta-param. r from μ,
  - residual error, reflecting deviation of observed score Y from what would be expected given the first three terms.

VCA in classical ANOVA [Fisher, 1925, Searle et al., 1992]

- Components in decomposition are uncorrelated with each other.
- Total variance $\sigma^2(Y-\mu)$ can be decomposed into following **variance components**:

$$\sigma^2(Y-\mu) = \sigma_s^2 + \sigma_r^2 + \sigma_{res}^2,$$

  - σ_s^2 and σ_r^2 denote variance due to sentences and meta-parameter settings,
  - σ_{res}^2 denotes residual variance including variance due to interaction of s and r.

Reminder: General Form of LMEMs

- For given dataset of N input-output pairs $\{(\mathrm{x}^n, y^n)\}_{n=1}^N$, general form of an LMEM is

$$\mathrm{Y} = \mathrm{X}\boldsymbol{\beta} + \mathrm{Zb} + \boldsymbol{\epsilon}.$$

  - Y are N stacked response variables,
  - X and Z known design matrices,
  - $\boldsymbol{\beta}$ fixed effects,
  - b random effects,
  - $\boldsymbol{\epsilon}$ residual errors,
  - where $\mathrm{b} \sim \mathcal{N}(0, \psi_\theta)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \Lambda_\theta)$.

- Conditions of measurement that contribute to variance in the measurement besides the objects of interest (here: sentences) are called *facets* of measurement (example: meta-parameters).
  - Each **facet-specific component** $\nu_f = \mu_f - \mu$ modeled as component b_f of **random effects** vector b,
  - corresponding **variance component** σ_f^2 modeled as component of **variance-covariance matrix** ψ_θ.

Advantages LMEM over ANOVA

- **Flexibility!**
  - General estimation procedure that is not design-driven.
  - Elegant handling of missing data situations.
  - Flexible modeling, e.g., random-effects-only models.
- Further reading: [Baayen et al., 2008, Barr et al., 2013, Bates et al., 2015]

Modeling Interactions with Data Properties in LMEMs

- Identify facet f with large variance contribution σ_f^2 in VCA.
- Analyze interaction of facet f with data property d:
  - Change random effect b_f to fixed effect β_f,
  - Add fixed effect β_d modeling test data characteristics,
  - Add interaction effect $\beta_{f:d}$ modeling interaction between data property d and facet f.

Intra-class correlation coefficient (ICC) [Fisher, 1925]

- Fundamental interpretation as measure of proportion of variance that is attributable to objects of measurement.
- Ratio of variance between objects of interest σ_B^2 to the total variance σ_{total}^2, including variance within objects of interest σ_W^2.

$$ICC = \frac{\sigma_B^2}{\sigma_{total}^2} = \frac{\sigma_B^2}{\sigma_B^2 + \sigma_W^2}.$$

- Name of coefficient is derived from goal of measuring how strongly objects in the same class are grouped together: **Variance between objects of interest should outweigh variance within!**

General reliability coefficient φ [Brennan, 2001]

- Ratio of substantial variance σ_s^2 to the sum of itself and absolute error variance σ_Δ^2, defined for facets $f_1, f_2, \ldots$ and selected interactions $s : f_1, s : f_2, f_1 : f_2, \ldots,$ all modeled as random effects:

$$\varphi = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_\Delta^2}, \text{ where } \sigma_\Delta^2 = \sigma_{f_1}^2 + \sigma_{f_2}^2 + \ldots + \sigma_{s:f_1}^2 + \sigma_{s:f_2}^2 + \ldots + \sigma_{f_1:f_2}^2 + \cdots + \sigma_{res}^2.$$

Reliability coefficient φ applied to NLP/data science

- **Reliability of performance evaluation across replicated measurements** is assessed as the **ratio by which the amount of substantial variance outweighs the total error variance**.
  - Variance should explained by variance between test sentences, not by variance-inducing facets like meta-parameter settings or by unspecified facets of measurement procedure.
  - Interpretation of threshold on ratio:
    - Values less than 50%, between 50% and 75%, between 75% and 90%, and above 90%, indicative of poor, moderate, good, and excellent reliability [Koo and Li, 2016]

Example: Variance Component Analysis of Meta-Parameter Importance

Assessing importance of meta-parameters

- Goal: Assess importance of meta-parameters in automatic meta-parameter search. [Habelitz and Keuper, 2020]
- Method: VCA using LMEM with random effects for meta-parameters (and interactions)
  - LMEMs offer unified framework to assess importance of meta-parameter across all levels of other meta-parameters, not just in context of a single fixed instantiation of remaining meta-parameters.
  - Previous work used less flexible functional ANOVA for same purpose. [Hutter et al., 2014, Zimmer et al., 2020]

Example: VCA of Meta-Parameter Importance

Example: A neural model for disease score prediction

- Multi-layer perceptron (MLP) to predict Sequential Organ Failure Assessment (SOFA) score.
- Meta-parameters:
  - maximal number of neurons in hidden layer (`hidden_size_max`),
  - number of hidden layers (`hidden_number`),
  - values of initial learning rate (`learning_rate`),
  - number of training examples in each gradient computation (`batch_size`),
  - seed of random number generator (`random_seed`),
  - number of iterations over training set (`epochs`),
  - probability of zeroing out hidden connections during training (`dropout`).

Example: VCA of Meta-Parameter Importance

Meta-parameter	Grid values					
batch_size	1	4	8	16	32	64
dropout	0	0.05	0.1	0.15	0.2	
epochs	1	5	10			
hidden_number	3	5	7			
hidden_size_max	16	32	64	128	256	
learning_rate	0.001	0.01	0.1			
random_seed	−7712	6483	20777			

- Meta-parameter values in grid search for SOFA-score MLP.

Example: VCA of Meta-Parameter Importance

- Random-effects-only LMEM:

$$Y = \mu + b_{hidden\_size\_max} + b_{hidden\_number} + b_{learning\_rate} + b_{batch\_size} + b_{random\_seed} + b_{epochs} + b_{dropout} + \epsilon_{res}.$$

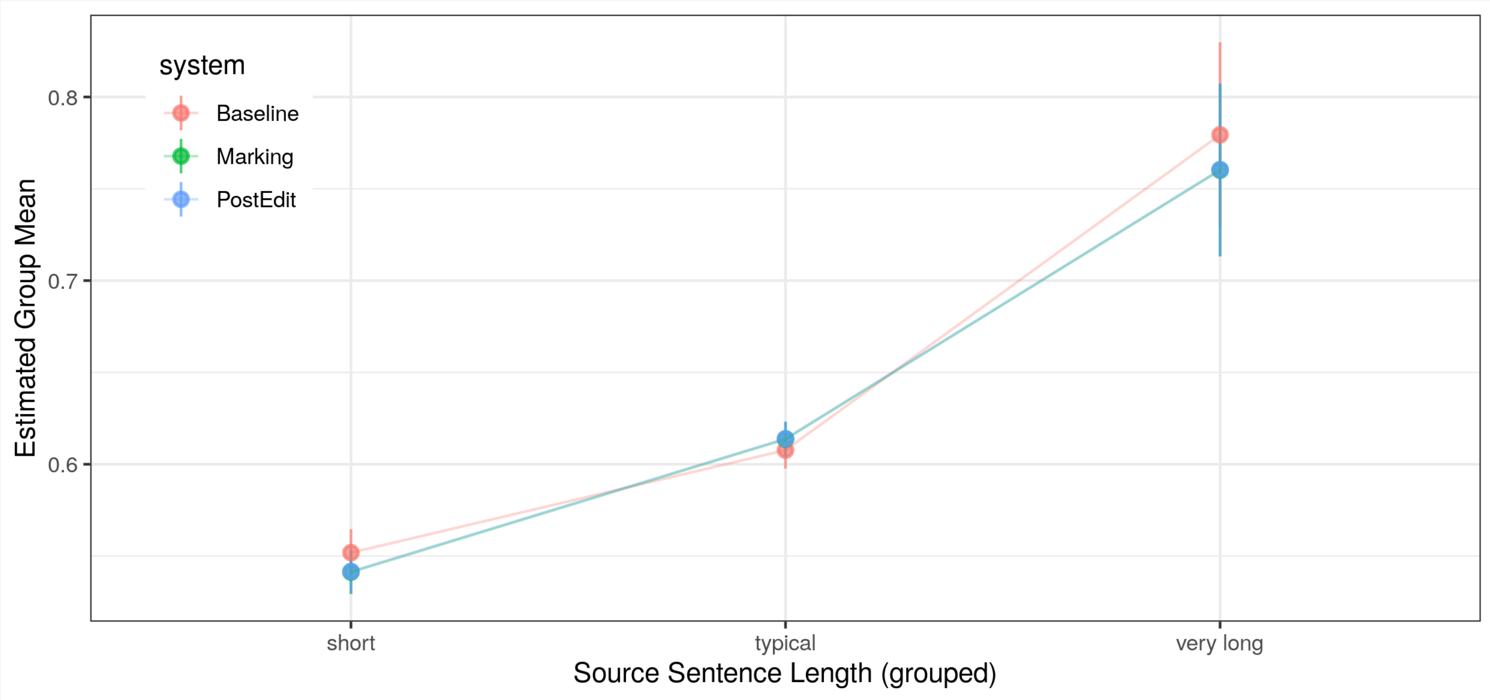
- Training data for LMEM:
  - Performance evaluations of summative evaluation metric, e.g., mean accuracy over test data instances.
  - Evaluations for fully crossed meta-parameter configuration space, yielding $6 \times 5 \times 3 \times 3 \times 5 \times 3 \times 3 = 12{,}150$ models.

Example: VCA of Meta-Parameter Importance

Variance component v	Variance σ_v^2	Percent
residual	0.0000314	61.2
hidden_number	0.0000159	31.0
learning_rate	0.00000318	6.2
batch_size	0.000000517	1.01
hidden_size_max	0.000000260	0.505
dropout	0.0000000599	0.117
random_seed	0.00000000405	0.00788

- Most variance induced by variation in number of hidden layers (31%),
- followed with a wide margin by learning rate (6.2% of total variance),
- all other meta-parameters introduce negligible variance of $\leq 1\%$.

Example: Reliability Analysis in Interactive Machine Translation



- Reminder: Significance between baseline and SOTA model was lost in extended meta-parameter grid search.
- Goal: Reliability analysis of SOTA model!
- Question: Which **meta-parameter setting is responsible** for performance drop, and what is **interaction with data** properties?

- Response variable Y is TER score on test sentence, μ is grand mean, b_s is sentence-specific deviation, and b_{random_seed} is random effect modeling 3 random seeds:

$$Y = \mu + b_s + b_{random\_seed} + \epsilon_{res}.$$

- Excellent reliability $\varphi = 98.4\%$, essentially no contribution of variance due to replications under random seeds.

Variance component	Variance σ^2	Percent
sentence	0.984	98.4
residual	0.0163	1.63
random_seed	0	0

Reliability Analysis of SOTA under Meta-Parameter Variation

- Add random effect b_f for each meta-parameter f in grid search:

$$Y = \mu + b_s + b_{learning\_rate} + b_{random\_seed} + b_{enc\_dropout} + b_{dec\_dropout} + b_{dec\_dropout\_h} + \epsilon_{res}.$$

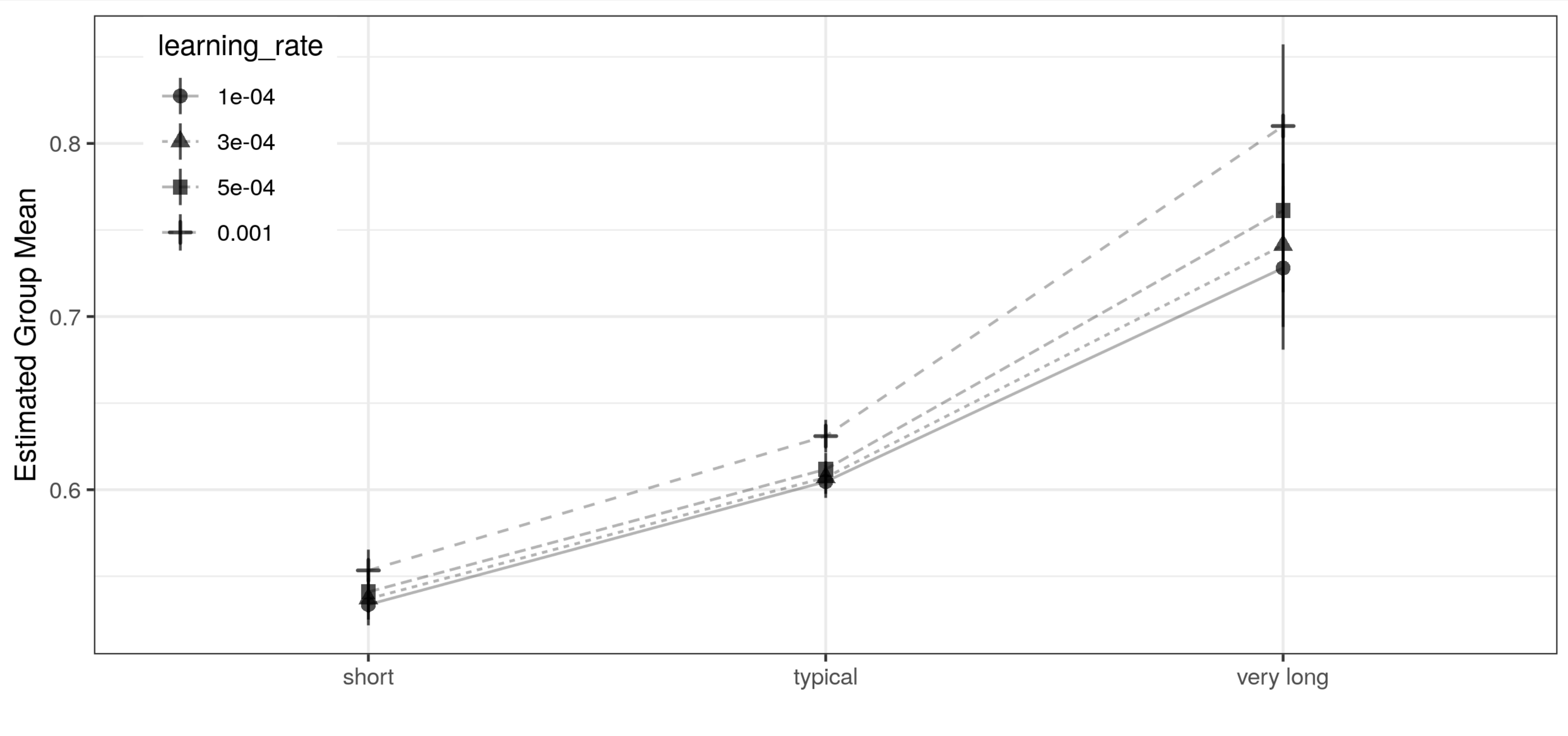
- Reliability coefficient drops below 90% with learning rate having largest contribution to variance.

Variance component	Variance σ^2	Percent
sentence	0.0574	88.4
residual	0.00737	11.3
learning_rate	0.000127	0.2
decoder_dropout	0.0000303	0.05
encoder_dropout	0.0000224	0.03
decoder_dropout_hidden	0.00000130	0
random_seed	0.000000578	0

Interaction between Meta-Parameters and Data Properties

- Add fixed effect β_{src_length} for source sentence length and interaction effect $\beta_{src\_length:learning\_rate}$.

$$Y = \mu + b_s + \beta_{src\_length} + \beta_{learning\_rate} + \beta_{src\_length:learning\_rate} + \epsilon_{res}.$$



- Significant improvements by fine-tuning over baseline with large effect size only on very long sentences.
  - ➡ Such improvements are likely to be reproducible on very long sentences of new datasets.
- Strong dependency of consistency of evaluation results on initial learning rate settings.
  - ➡ Likely that the results will be reproducible only for small initial learning rates (< 0.0005), but not for large initial learning rates.
- Questionable reproducibility of result differences on short and medium length sentences, especially between fine-tuned systems.

Summary: LMEM-Based Reliability Analysis

- Distinctive idea:
  - Compute **reliability coefficient as proportion of substantial variance attributable to the objects of interest**, compared to insubstantial variance due to idiosyncrasies of measurement situation.
  - Ideas date back to [Fisher, 1925] and allow **interpretation of reasons for (un)reliability** and **understanding of interactions of variance components and data**.
  - Based on **well-understood statistical models (LMEMs)**.
  - Further reading: [Searle et al., 1992, Brennan, 2001, Webb et al., 2006].

Agreement coefficients for data annotation

- Scott's π [Scott, 1955], Cohen's κ [Cohen, 1960], or Krippendorff's α [Krippendorff, 2004] are commonly used descriptive statistics to measure agreement of raters in data annotation.
- Based on simple concept of percent agreement that is adjusted to include agreement by chance.
- Easily computable from experimental data by collecting relative count statistics.

Problems with agreement coefficients

- Convenience in computation is due to a fixed choice of a model for computing chance agreement:
  - Sampling with replacement (Scott's π and Cohen's κ) or without replacement (Krippendorff's α),
  - from distributions for individual raters (Cohen's κ) or from observed ratings averaged over raters (Scott's π and Krippendorff's α).

Problems with agreement coefficients

$$\text{chance-adjusted agreement} = \frac{\text{observed agreement - chance agreement}}{n - \text{chance agreement}}.$$

- Counter-intuitive principle of maximum randomness, leading to many paradoxes and abnormalities. [Zhao et al., 2013]
- Main disadvantages:
  - No generalization beyond concrete raters and concrete data points examined in a concrete experiment.
  - No explanation of reasons for high/low agreement by properties of raters or data, or by interactions between them.

Bootstrap confidence intervals for model evaluation

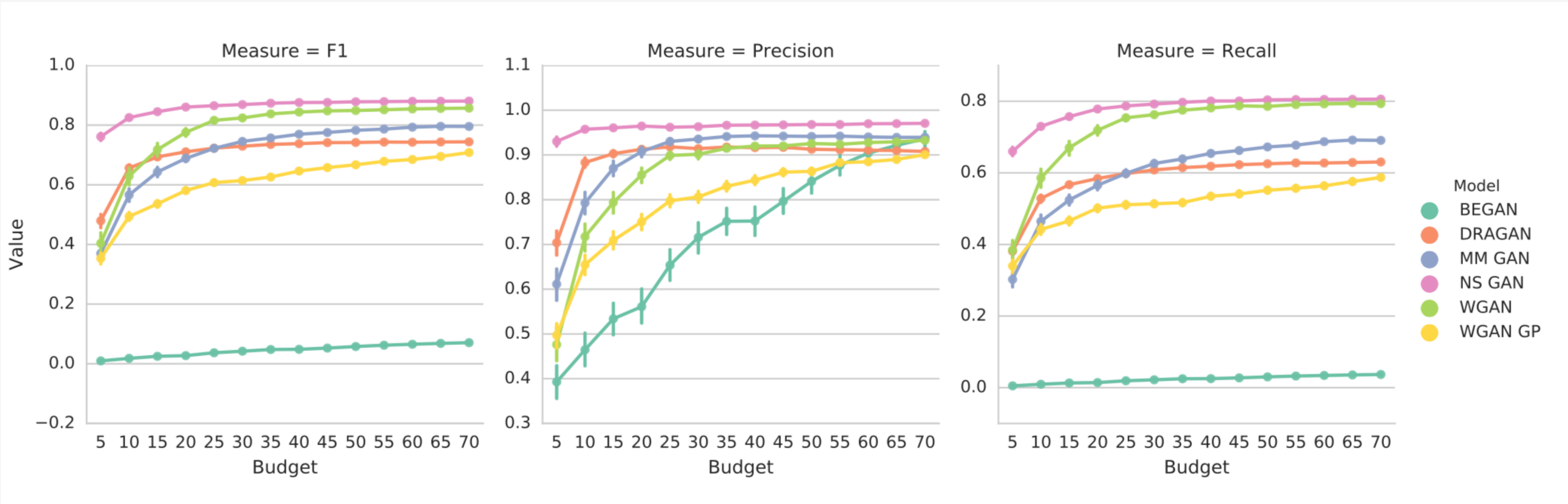
- Interest is in reliability of predictions of a machine learning algorithm itself, not just reliability of single concrete evaluation experiment.
- Bootstrap-inspired resampling to compute confidence bounds for evaluation scores on test data. [Henderson et al., 2018, Lucic et al., 2018]
  - Goal: Quantify variation in maximum out-of-sample performance with respect to meta-parameter choice and computational budget.
  - Method: Resample performance evaluation scores from pool of models trained under increasing budget for meta-parameter search.

Bootstrap Confidence Interval for Evaluation Metric

1. Generate M meta-parameter configurations for the model class.
2. For each $m = 1, \ldots, M$: Train model p_m and calculate the performance evaluation score $u_m = u(p_m)$.
3. For each $B \leq M$: Construct a bootstrap distribution by K times drawing B random samples with replacement from $\{u_m \colon m = 1, \ldots, M\}$. For each sample select the maximum performance score.
4. Calculate the mean $\bar{x}$ and the standard deviation $\sigma_{\bar{x}}$ of this distribution. Plug both estimates into the standard normal 95% confidence interval of the population mean μ:

$$\bar{x} - 1.96\sigma_{\bar{x}} \leq \mu \leq \bar{x} + 1.96\sigma_{\bar{x}}.$$

- Mean and 95% confidence intervals for F1-score, precision, recall of GANs for different computational budgets. [Lucic et al., 2018]



Problems with bootstrap confidence intervals

- Idea: Use confidence bounds to directly signify reliability of an evaluation meta-parameter settings: At the same level of confidence, smaller confidence bounds indicate higher reliability.
- Problems:
  - Lacking bootstrap consistency, either if test set from which bootstrap samples are drawn is not representative of population [Canty et al., 2006], or if the parameter to be estimated is on the boundary of the parameter space [Andrews, 2000, Bickel and Freedman, 1981] as in calculations of expected maximum performance [Lucic et al., 2018, Dodge et al., 2019].
- Main shortcoming:
  - Confidence intervals do not tell us which meta-parameters have the most influence on variations in evaluation scores, and how meta-parameter settings interact with properties of test data.