Lecture 3: Inductive bias, regularization, and ensemble methods

Stefan Bucher

Stefan Bucher

Open in Colab

Prince (2023, chaps. 5, 9); Murphy (2022, chap. 18).[1]

. Figures taken or adapted from Prince (2023). All rights belong to the original author and publisher. These materials are intended solely for educational purposes.

Generalization and inductive bias

- **Generalization:** performance on unseen (test) data
- **Inductive bias:** assumptions that favour some solutions over others (e.g. smoothness, sparsity); determines model behavior in underdetermined regimes and hence generalization.
- Today: concrete tools to improve generalization.

Outline

1. **Regularization** — constrain or perturb parametric models (L2, L1, dropout, transfer, etc.).
2. **Nonparametric methods** — Classification and Regression Trees (CART).
3. **Ensemble methods** — Bagging, random forests, boosting, XGBoost.
4. **Interpretability** — Shapley values and SHAP for feature attribution.

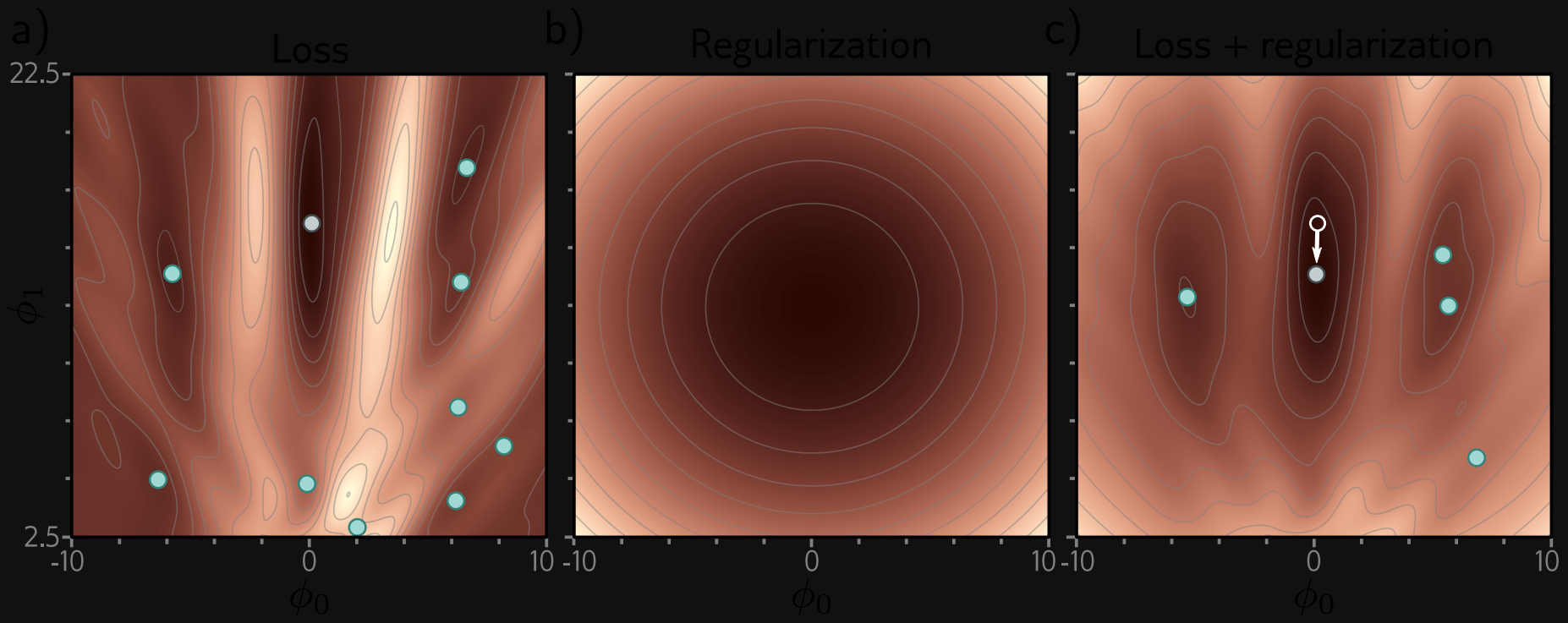
Regularization

Goal: Reduce generalization gap between training and test performance.

Prince (2023, chap. 9)

Explicit Regularization

$$\hat{\phi} = \underset{\phi}{\arg\min} \sum_{i=1}^{I} l_i[x_i, y_i] + \lambda g[\phi]$$



Interpretation as Bayesian Prior

Maximum a posteriori (MAP) instead of maximum LLH

$$\hat{\phi} = \arg\max_{\phi} \prod_{i=1}^{I} q(y_i|\theta_i = f[x_i, \phi])P(\phi)$$

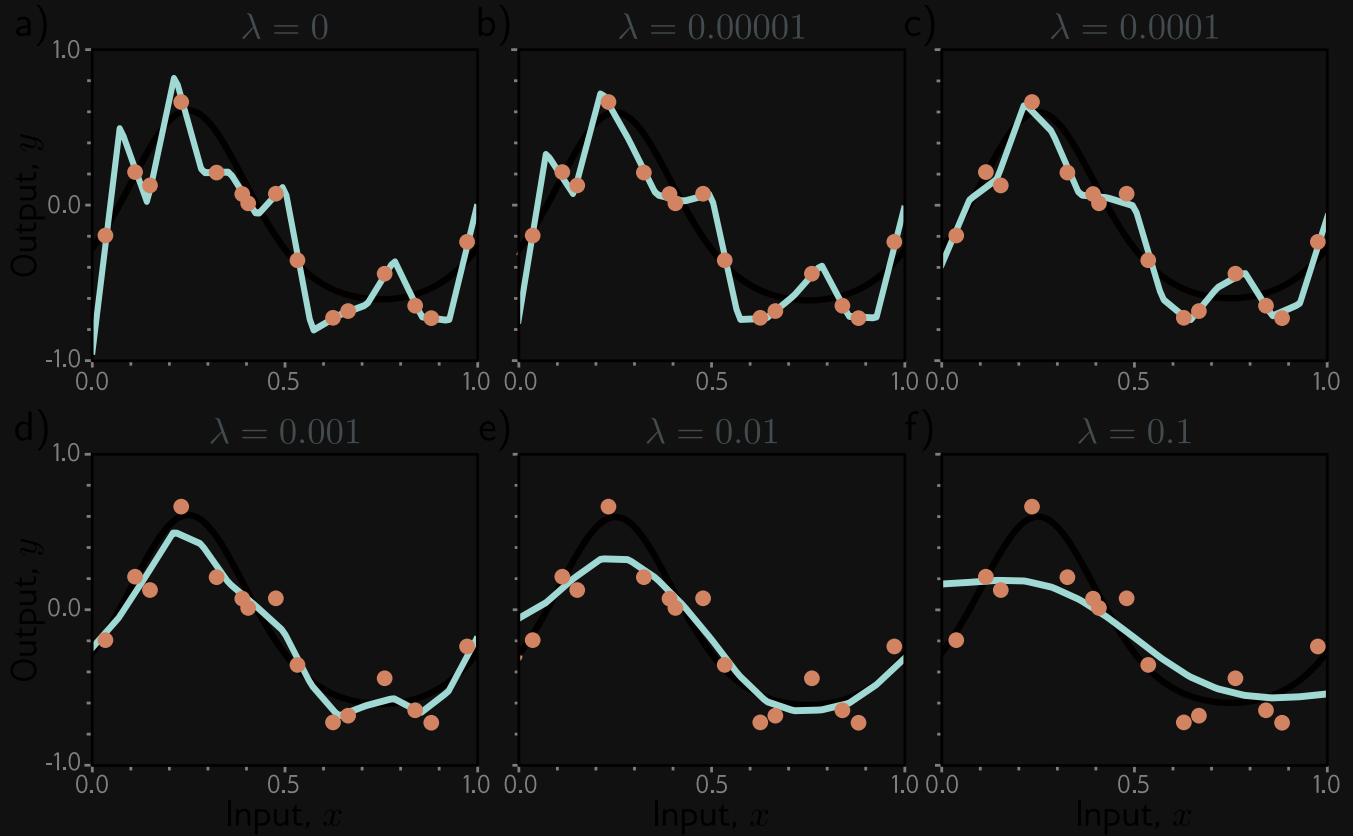
$$= \arg\max_{\phi} \log \prod_{i=1}^{I} q(y_i|\theta_i = f[x_i, \phi])P(\phi)$$

$$= \arg\min_{\phi} -\log P(\phi) - \sum_{i=1}^{I} \log q(y_i|\theta_i = f[x_i, \phi])$$

$$= \arg\min_{\phi} \lambda g[\phi] + \sum_{i=1}^{I} l_i[x_i, y_i]$$

L2 regularization: Ridge regression

$$g[\phi] = \sum_j \phi_j^2$$



L1 regularization: Lasso regression

$$g[\phi] = \sum_j |\phi_j|$$

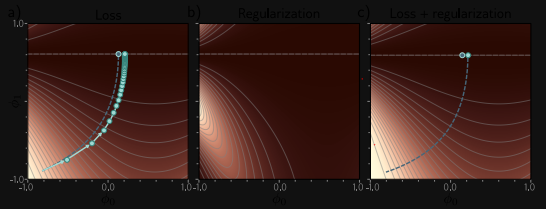
Implicit Regularization

Discrete stochastic gradient descent on L arrives at same place as continuous descent $\frac{d\phi}{dt} = -\frac{\partial \tilde{L}}{\partial \phi}$ on

$$\tilde{L}[\phi] = L[\phi] + \frac{\alpha}{4}||\frac{\partial L}{\partial \phi}||^2 + \frac{\alpha}{4B}\sum_{b=1}^{B}||\frac{\partial L_b}{\partial \phi} - \frac{\partial L}{\partial \phi}||^2$$

so is naturally biased

- away from steep gradients
- to stable gradients (low variance across batches), suggesting that “all data fits well” and promising better generalization



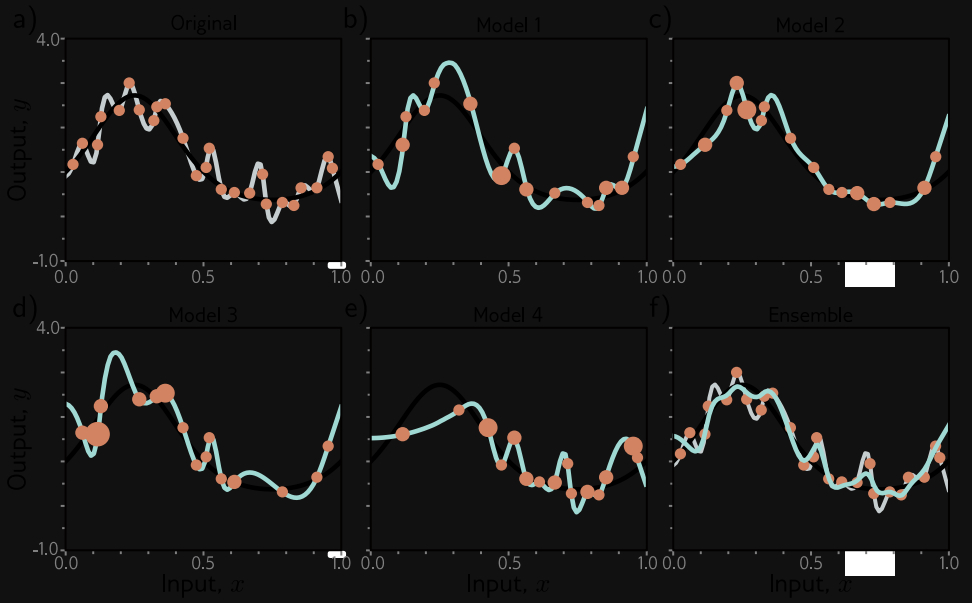
Performance Heuristics

- Early stopping (to avoid overfitting)
- Applying noise (or adversarial) during training (to inputs, weights, or labels) for increased robustness
- Bayesian inference $P(\phi|\{x_i, y_i\}) = \propto \prod_{i=1}^{I} q(y_i|x_i, \phi)P(\phi)$ (often not practical)
- Normalization

Ensemble Methods

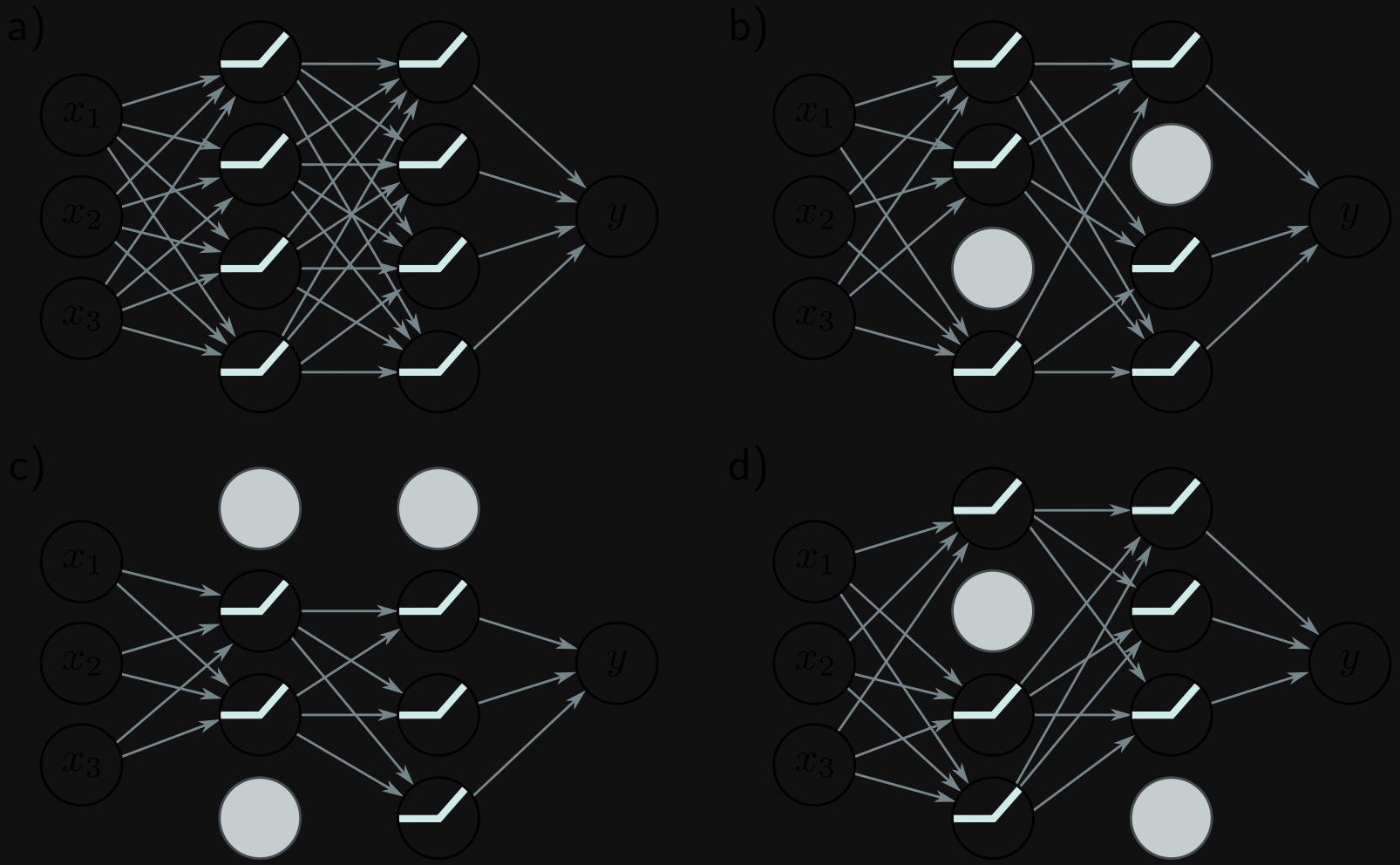
Average predictions of multiple models, e.g.

- different initializations
- Bagging (bootstrap aggregating, i.e. resampling training data)



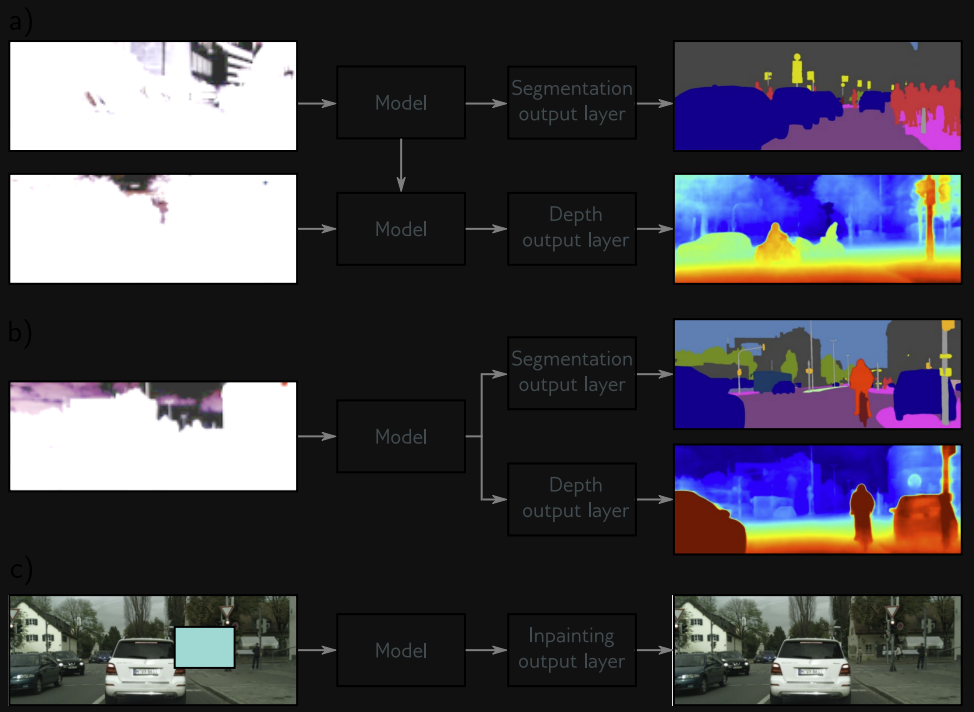
Dropout

Clamping subset of hidden units to zero.

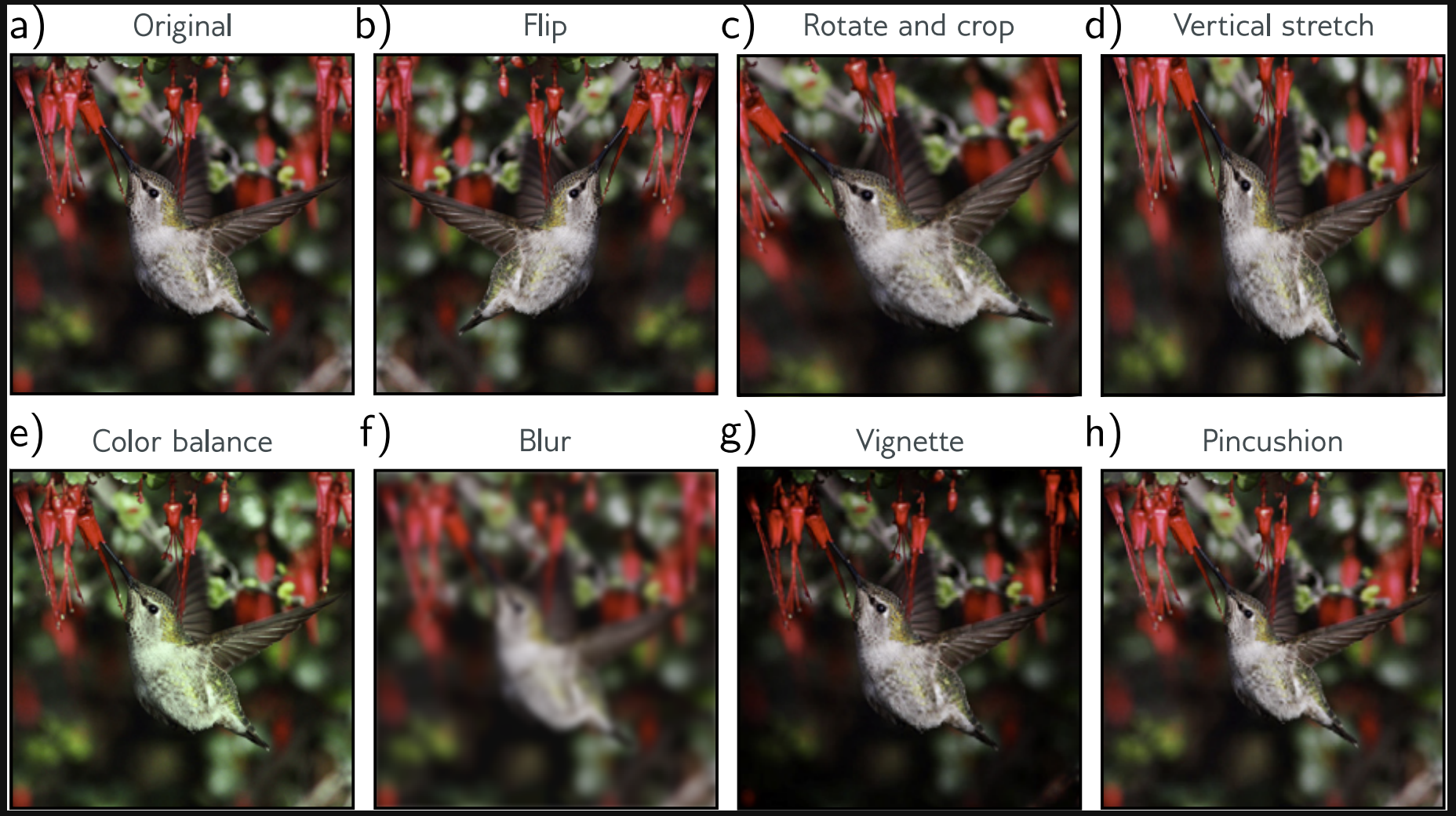


Transfer, Multi-task & Self-supervised Learning

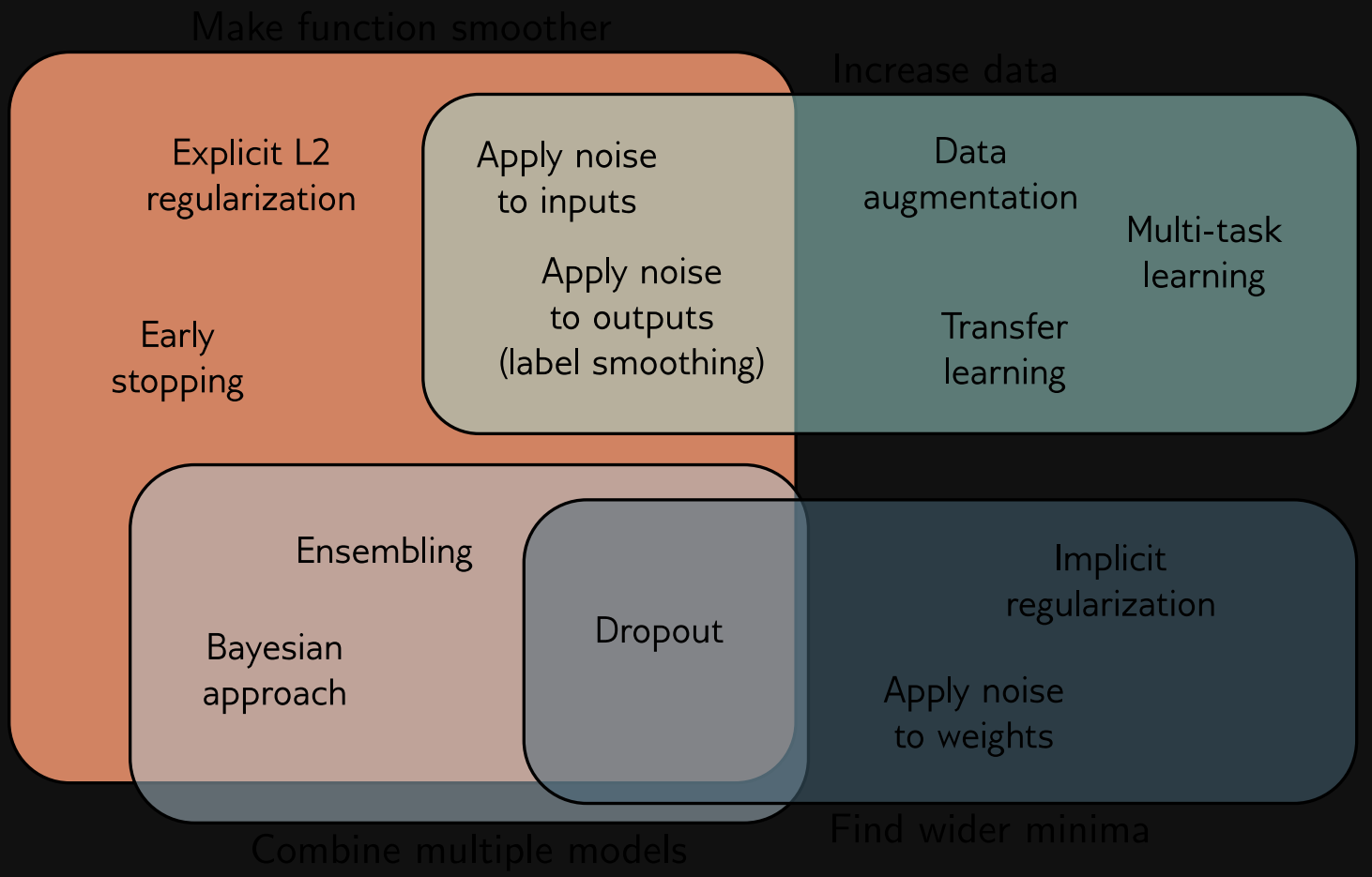
- Transfer learning (pretrain on related training data)
  - train final layers
  - fine-tune whole model
- Multi-task learning
- Self-supervised learning
  - generative (fill in masked data)
  - contrastive (compare pairs of examples for relatedness)


a)
Model
Segmentation output layer
Model
Depth output layer
b)
Model
Segmentation output layer
Depth output layer
c)
Model
Inpainting output layer

Data Augmentation



Regularization Overview



Nonparametric methods: CART

Murphy (2022, chap. 18.1)

CART: Classification and Regression Trees

- **Recursive partitioning:** choose split (feature + threshold) to minimize impurity
- regression: MSE
- classification: entropy or Gini
- **Axis-aligned splits**: input space partitioned into **regions** (axis-aligned boxes); prediction is **constant** in each region (one value per leaf).

Outlook: Neural networks also partition the input space into regions, but with **learned** (non–axis-aligned) boundaries and piecewise linear behavior.

Ensemble methods

Murphy (2022, chaps. 18.2–18.5)

Explained here through trees; the ideas (averaging, sequential correction) apply to other base learners too.

The Challenge with Trees

- interpretable, but high variance (sensitive to data).

Bagging

- **Bootstrap aggregating:** fit many models (e.g. trees) on bootstrap samples of the training data; average their predictions.
- **Variance reduction** by averaging.

Random forests

- Bagging + at each split use a random subset of features (further decorrelates trees).
- Average or vote.

Boosting

- **Sequential:** each new model focuses on errors (residuals) of the previous ones.

Gradient boosting and XGBoost

- **Gradient boosting:** fit each tree to the negative gradient of the loss.
- **XGBoost:** efficient implementation, widely used for tabular data.

Model interpretability

Shapley values and SHAP

- **Shapley value (game theory):** Each player's marginal contribution to a collective payoff, averaged over all possible permutations of joining a coalition.
  - The only "fair" allocation method from cooperative game theory satisfying axioms like efficiency and symmetry.
- **SHAP (SHapley Additive exPlanations, Lundberg & Lee, 2017):** Applies this theory to models by treating features as players and the prediction as the payoff; provides additive feature attributions that are model-agnostic, consistent, and locally accurate.