# Problem setting

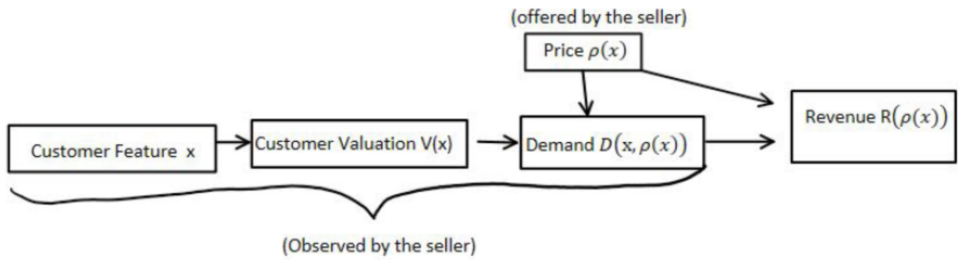



Figure: Pricing Framework

## Problem Formulation

Goal: find a policy $\rho$ that maximizes the seller's revenue (R) while minimizing the prediction error of learning the demand function,

$$\min_{\rho,d,f} \mathbb{E}_{(\mathbf{x},p,\tilde{d})\sim D_T} \|d(f(\mathbf{x}),p;\theta_d)-\tilde{d}\|_2-\lambda\, d(f(\mathbf{x}),\rho(f(\mathbf{x}));\theta_d)\rho(f(\mathbf{x})), \tag{1}$$

where

- $\mathbf{x}$, $\tilde{d}$, $p$ are observed from the data
- $d(\cdot;\theta_d) : D \to \mathbb{R}_+$ is the demand function parametrized by $\theta_d$ that satisfies assumption 1.
- $f(\cdot) : D \to D_z$ is a feature extractor
- $\lambda$ is a hyper-parameter

### Assumption 1 (The law of demand)

*Quantity purchased varies inversely with price.*

## Problem Formulation

In model training, the objective is defined as,

$$\min_{\rho,\theta_d,f,g} \quad \mathbb{E}_{(\mathbf{x},p,\tilde{d})\sim D_T\cup D_S} \|d(f(\mathbf{x}),p;\theta_d)-\tilde{d}||_2$$
$$-\lambda\, d(f(\mathbf{x}),\rho(f(\mathbf{x}));\theta_d)\rho(f(\mathbf{x}))$$
$$-\gamma\,(y\log(h(f(\mathbf{x})))+(1-y)\log(1-h(f(\mathbf{x})))), \tag{2}$$

where

- $h(\cdot)$ is the domain classifier
- $y = 1(x \in D_S)$ is the domain label

## JPO Loss

### Definition 1

(JPO Loss). Given a vector of observed information $(\mathbf{x}, \tilde{d}, p)$, the JPO loss $L_{\lambda_0}^{JPO}$ with respect to a hyper-parameter $\lambda = \lambda_0$, where $\lambda_0$ could guarantee convergence is defined as $\mathbb{E}_{(\mathbf{x},p,\tilde{d})\sim D_T} \|d(f(\mathbf{x}),p;\theta_d)-\tilde{d}||_2-\lambda_0\, d(f(\mathbf{x}),\rho(f(\mathbf{x}));\theta_d)\rho(f(\mathbf{x}))$ , $\forall d \in \Gamma_d$ , where $\Gamma_d$ is a restrictive function class that only contains functions satisfying assumption 1 (i.e., monotonic with respect to $p$ and $\rho$).

- JPO loss is adopted in UMNN algorithm

# JPO+ Loss

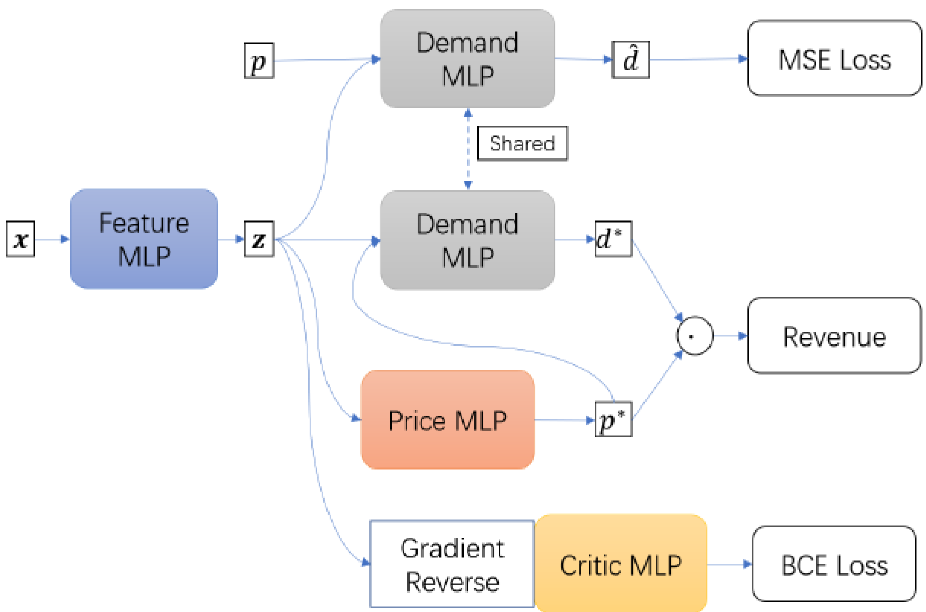
- If function $d$ is differentiable,

**Definition 2**

(JPO+ Loss). Given a vector of observed information $(\mathbf{x}, \tilde{d}, p)$, the JPO+ loss $L_{\lambda}^{JPO+}$ with respect to any hyper-parameter tuples $\lambda = (\lambda_0, \lambda_1, \lambda_2)$ is defined as $\mathbb{E}_{(\mathbf{x},p,\tilde{d})\sim D_T} \|d(f(\mathbf{x}), p; \theta_d) - \tilde{d}\|_2^2 - \lambda_0\, d(f(\mathbf{x}), \rho(f(\mathbf{x})); \theta_d)\rho(f(\mathbf{x})) + \lambda_1 \frac{\partial d}{\partial p} + \lambda_2 \frac{\partial d}{\partial \rho}$

- JPO+ loss is a Lagurangian relaxation of JPO loss.
- JPO+ loss is used in MLP algorithm

# Model Architecture



# Experiment Setup

Synthetic Data Setting

- Price values are sampled from the same uniform distribution for both source and target domains.
- Source features are sampled from multivariate distribution while target features are sample from chi-squared distribution.
- Demand function is linear w.r.t to features and price.
- Half of the features are dropped out in the target domain.
- Source domain has 1M data samples, while target domain only has 1K samples.

# Experiment Results

- Enforcing monotonicity can drastically increase the R/MSE ratio.
- Domain Adaption loss improves domain generalization.

| Model | Revenue (R) | MSE | R/MSE |
|---|---|---|---|
| LR baseline | 0.479 | 0.093 | 5.15 |
| MLP w/o DA | 0.861 | 0.090 | 9.57 |
| MLP | 0.794 | 0.066 | 12.03 |
| MLP w/ GradReg | 0.565 | 0.023 | 24.57 |
| UMNN | 0.619 | 0.019 | **32.58** |