Generalized and Optimal Straight-Through Estimators

James David Hooper **Alexander Shekhovtsov**
Czech Technical University in Prague

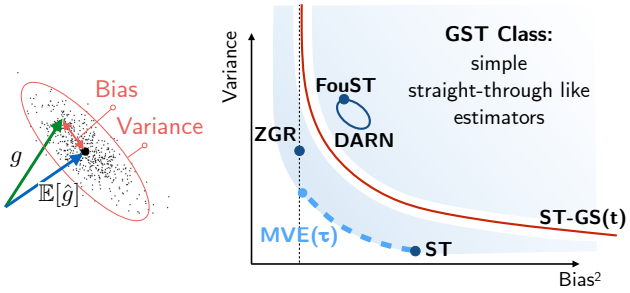
Abstract

Modern ML models often utilize discrete components within their computational graphs, making training challenging. In such cases, approximate-chain-rule gradient estimators can be applied. They work reasonably well but are obtained by combining diverse rationales with ad-hoc choices. In this work, we propose a principled axiomatic approach to define a general family of gradient estimators and show that it subsumes many existing methods. Within this family, we derive optimal estimators with respect to a minimum variance criterion subject to interpretable bias-limiting constraints, addressing integer and one-hot categorical discrete variables. We empirically demonstrate that our estimator can achieve a better bias-variance trade-off than existing ones on synthetic problems and outperforms them on training variational auto-encoders with discrete latent variables.

1 INTRODUCTION

Discrete operations and variables are essential in modern machine learning. Examples include quantized weights and activations: binary (Hubara et al., 2017; Lin et al., 2017), integer (Louizos et al., 2019), and vector-quantized (Savkin et al., 2025), sparsity models (Jayakumar et al., 2020), discrete representations in variational autoencoders (Jang et al., 2017; Maddison et al., 2017; van den Oord et al., 2017; Khalil et al., 2023), discrete world models and action spaces in reinforcement learning (Hafner et al., 2021), and more.

Modern machine learning succeeded in scaling to large datasets and complex models thanks to stochastic gradient optimization, which optimizes the expected loss by using the gradient evaluated at a data sample. Handling discreteness in this framework is possible in a principled manner when the discrete variables are random and the loss is differentiable in expectation. In such situations, stochastic estimates of the gradient can be computed using a single sample of discrete variables. In a general setting, score-based unbiased gradient estimators can be applied (e.g., Kool et al., 2019); they estimate the gradient exclusively from the values of the loss function, avoiding backpropagation entirely. However, in many practical cases of interest, approximate-chain-rule estimators, which substitute surrogate Jacobians during backpropagation, perform better due to their lower variance. This variance reduction comes at the cost of introducing bias, as illustrated in Fig. 1 The tradeoff is, however, still beneficial and often crucial for many applications.



(a) (b)

Figure 1: (a) Bias and variance of a stochastic gradient estimator $\hat{g}$ with respect to the true gradient g of the expected loss. (b) Illustration of the bias-variance trade-off for different estimators. The proposed general class contains ST, FouST, and ZGR and we aim to optimize the bias-variance trade-off within this class.

Existing approximate-chain-rule estimators, such as straight-through (ST), Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017), and their variants (Pervez et al., 2020; Paulus et al., 2021; Shekhovtsov, 2023), discussed below, were derived from different conceptual approaches, often relying on heuristic choices. This raises the question of whether there is

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

a more principled approach to derive such estimators and whether better estimators can be found within a broader class.

In this work, we propose a principled axiomatic approach: we postulate a few elementary properties that good estimators must satisfy, and obtain a class of simple straight-through-like estimators. We show that ST, FouST, DARN Gregor et al. (2014) and ZGR are members of this class. We then aim to identify the Pareto-optimal frontier of this class as illustrated in Fig. 1. Towards this end, we consider several proxy variance and bias measures, which do not depend on the unknown loss function, and solve for the minimum variance estimators under a constrained bias. This yields new derivations of ST and ZGR in the general categorical case as minimum variance estimators (MVEs) subject to different constraints. By varying the Lagrange multiplier of the bias constraints, we obtain a 1-parameter family of optimal estimators with respect to the bias-variance trade-off (in the sense of the proxy measures). In our experiments, we observe that our minimum variance parametric family indeed spans a range of bias-variance trade-offs that are superior to those of existing estimators. For quantized networks, it outperforms ZGR, and for discrete VAEs, it outperforms all existing estimators when using a simple schedule transitioning from the low variance to the low bias regime during training.

2 BACKGROUND AND RELATED WORK

Approximate-chain-rule estimators are typically defined locally for a single discrete random variable, assuming all other random variables are fixed. Let $x \in \mathcal{X}$, where $|\mathcal{X}| = K$, be a discrete random variable with distribution $p(x;\eta)$ parametrized by $\eta \in \mathbb{R}^r$. Let $\phi(x) \in \mathbb{R}^d$ be an *embedding* of the discrete state x, enabling its use in downstream algebraic expressions. Let $\mathcal{L}(\phi(x))$ be a real-valued loss function, differentiable in ϕ. Where appropriate, we identify x with $\phi(x)$, *i.e.*, $\mathcal{L}(x) \equiv \mathcal{L}(\phi(x))$.

The formal problem studied in this work is estimating the gradient of the expected loss with respect to the parameters η:

$$J_\eta = \frac{\mathrm{d}}{\mathrm{d}\eta}\mathbb{E}_x[\mathcal{L}(x)] = \frac{\mathrm{d}}{\mathrm{d}\eta}\sum_x p(x;\eta)\mathcal{L}(x). \tag{1}$$

In some applications, this problem arises naturally, *e.g.*, in discrete VAEs, where p is the encoder distribution. In other applications, it originates from a stochastic relaxation, which replaces discrete optimization with continuous optimization over the parameters of the distribution of the discrete variables (*e.g.*, training NNs with quantized weights, probabilistic subsampling, Huijben et al. 2020).

For a small space $\mathcal{X}$, it is straightforward to compute (1) by enumerating over $x \in \mathcal{X}$. Instead, we are interested in a stochastic estimator of J_η using a single sample $x \sim p(x;\eta)$, which is useful when $\mathcal{X}$ is large or when there are many such variables in the model. The naive chain rule $\frac{\mathrm{d}\mathcal{L}(\phi)}{\mathrm{d}\phi}\frac{\mathrm{d}\phi(x)}{\mathrm{d}\eta}$ is clearly not applicable, as $\phi(x)$ does not depend on η, and it fails to account for the dependence of the expectation $\mathbb{E}_x$ on η. To address this, a reparameterization $x = g(\eta, Z)$ can be applied, which allows us to replace E_x with E_Z, where the distribution of Z does not depend on the parameters η. However, E_Z and $\frac{d}{d\eta}$ still cannot be interchanged because g is necessarily discrete in η. For any sample Z, the derivative of $\mathcal{L}(\phi(g(\eta, Z)))$ evaluates to zero, offering no utility in estimating the derivative of the expectation (1).

Among the most common approximate-chain-rule estimators, there is the straight-through estimator, colloquially proposed by Hinton and formalized much later (Tokui and Sato 2017; Shekhovtsov and Yanush 2021; Liu et al. 2023). Let us define the *mean embedding* $\bar{\phi}(\eta) = \mathbb{E}[\phi(x)] = \sum_x p(x;\eta)\phi(x)$.

Straight-Through (ST) estimator fixes the chain rule by using the derivative of the mean embedding $\bar{\phi}(\eta)$ instead of the non-existing derivative of $\phi(x)$:

$$\hat{J}_\eta^{\mathrm{ST}} = \frac{\mathrm{d}\mathcal{L}(\phi)}{\mathrm{d}\phi}\frac{\mathrm{d}\bar{\phi}(\eta)}{\mathrm{d}\eta}, \quad x \sim p(x;\eta),\ \phi = \phi(x). \tag{2}$$

Note that Bengio et al. (2013), often referenced for ST, substitute 1 instead of $\frac{\mathrm{d}\bar{\phi}(\eta)}{\mathrm{d}\eta}$. Furthermore, there are other popular empirical forms outside the context of the well-defined formulation (1), in particular in binary neural networks (Courbariaux and Bengio, 2016; Helwegen et al., 2019).

DARN/FouST (Gregor et al., 2014; Pervez et al., 2020) is an estimator with a free "baseline" parameter $\tilde{\phi}$ defined as follows:

$$\hat{J}_\eta^{\mathrm{DARN}(\tilde{\phi})} = \frac{\mathrm{d}\mathcal{L}(\phi)}{\mathrm{d}\phi}(\phi(x) - \tilde{\phi})\frac{d\log p(x;\eta)}{d\eta}. \tag{3}$$

For binary variables, *i.e.*, $\phi(x) \in \{0, 1\}$, Gregor et al. (2014) used $\tilde{\phi} = \frac{1}{2}$, which ensures that the estimator is unbiased for any quadratic function. It was later identified by Pervez et al. (2020) with a reweighting scheme applied to the empirical ST estimator, which can be written as:

$$\hat{J}_\eta^{\mathrm{FouST}} = \frac{\mathrm{d}\mathcal{L}(\phi)}{\mathrm{d}\phi}\frac{\mathrm{d}\bar{\phi}(\eta)}{\mathrm{d}\eta}\frac{1}{Kp(x;\eta)}. \tag{4}$$

More specifically, for binary variables, it matches $\mathrm{DARN}(\frac{1}{2})$ (Pervez et al., 2020). When generalizing to $K > 2$, however, the two constructs diverge and neither leads to good estimators in practice.

Gumbel-Softmax Family One of the most common estimators is the Gumbel-Softmax (GS) estimator (Jang et al., 2017; Maddison et al., 2017). It uses a smooth approximation of argmax with a tempered softmax to define a differentiable reparameterization. The vanilla GS estimator computes the forward pass using relaxed (non-discrete) states. This often negatively impacts training, as all expectations become biased. The ST Gumbel-Softmax (STGS) estimator (Jang et al., 2017) is a popular variant, which uses discrete samples in the forward pass and the Jacobian of the relaxed reparameterization during the backward pass.

Subsequent works have proposed principled improvements to this approach: a provable variance reduction scheme for **STGS** (Paulus et al., 2021), denoted **GRMC**, though it requires Monte Carlo sampling inside the gradient estimator; and the zero-temperature limit of the variance-reduced estimator, denoted **ZGR** (Shekhovtsov, 2023). Interestingly, the latter does not depend on the Gumbel distribution and is instead expressed as a simple closed-form combination of ST and DARN:

$$\hat{J}_\eta^{\mathrm{ZGR}} = \tfrac{1}{2}\Big(\hat{J}_\eta^{\mathrm{ST}} + \hat{J}_\eta^{\mathrm{DARN}(\bar{\phi}(\eta))}\Big). \tag{5}$$

For binary variables, it coincides with DARN($\frac{1}{2}$), and in the categorical case, it is unbiased for any quadratic function, making it a strong generalization of the DARN design.

ReinMax Independently from the above developement, Liu et al. (2023) have proposed an estimator, motivated by Heun's method for numerical ODEs, approximating finite differences $\mathcal{L}(x)-\mathcal{L}(x')$ with second order accuracy. Both ReinMax and ZGR were demonstrated to outperform other biased and unbiased methods in training discrete VAEs and other problems.

To summarize, we have seen that ST, FouST, DARN, ReinMax and Gumbel-Softmax variants are derived from different principles, often relying on heuristic choices in the construction / generalization to the categorical case. ZGR and ReinMax appear to combine severable desirable properties, but the question remains whether there is a common rationale for them and whether they are optimal.

3 GENERALIZED STRAIGHT-THROUGH

We propose an axiomatic approach to characterize a broad class of estimators which can be used as plug-in replacements for the chain rule in backpropagation through discrete variables. We call them *Generalized Straight-Through* (GST) estimators. Let $\pi \in \Delta^{K-1}$ denote the vector of probabilities $\pi_k = p(x{=}k;\eta)$ for $k = 1 \dots K$ and let us denote the Jacobian of π w.r.t. η as: $(J_\eta^\pi)_{k,j} = \frac{\mathrm{d}p(x=k;\eta)}{\mathrm{d}\eta_j}$.

Definition 1 (*Generic Estimator*)**.** *We define a generic estimator as a method that, for any loss function $\mathcal{L}\colon \mathbb{R}^d \to \mathbb{R}$ and a given probability model $\pi\colon \mathbb{R}^r \to \Delta^{K-1}$, provides a potentially stochastic estimate of the derivative in π, denoted as $\hat{J}_\pi[\mathcal{L}, \pi(\eta), \omega]$, where ω is a random event capturing all sources of stochasticity. The estimate of the derivative in η is defined by the chain rule: $\hat{J}_\eta = \hat{J}_\pi J_\eta^\pi$.*

Axiom 1 (Linearity w.r.t. loss)**.** *Let $\hat{J}_\eta[\mathcal{L}]$ denote the estimator as a function of the loss $\mathcal{L}$. The estimator is linear w.r.t. the loss if it satisfies*

$$\hat{J}_\eta[\mathcal{L}_1 + \mathcal{L}_2] = \hat{J}_\eta[\mathcal{L}_1] + \hat{J}_\eta[\mathcal{L}_2] \tag{6}$$

for any $\mathcal{L}_1, \mathcal{L}_2\colon \mathbb{R}^d \to \mathbb{R}$, $\pi \in \Delta^{K-1}$ and ω.

All common estimators are linear in the above sense - they depend linearly either on the value of the function $\mathcal{L}$ at some point (for the REINFORCE family) or on its derivative. This property is essential in any interchange of the estimator with expectations, in particular when extending the estimator to multiple variables and applying it within stochastic gradient descent.

Axiom 2 (Unbiased for linear losses)**.** *If the loss function is linear $\mathcal{L}(x) = a^\mathsf{T}\phi(x)$, we posit that the expectation of the estimator $\mathbb{E}_\omega\big[\hat{J}_\eta\big]$ must match the true derivative $a^\mathsf{T}\frac{\mathrm{d}}{\mathrm{d}\eta}\mathbb{E}_x[\phi(x)]$.*

For estimators based on the derivative $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\phi}$, this captures the intuition that for smooth, and especially linear functions, this derivative is informative and useful for estimating J_η. ST, FouST, ZGR and ReinMax all satisfy this property, while estimators in the GS family (GS, STGS, GRMC) satisfy it only asymptotically for the temperature $t \to 0$.

Let $\Phi \in \mathbb{R}^{d\times K}$ be the *embedding matrix*, with components $\Phi_{j,k} = \phi(k)_j$, $k \in \mathcal{X}$.

Theorem 1 (Generalized ST)**.** *An estimator in the sense of Definition 1 that depends on $\mathcal{L}$ only through the derivative $J_\phi = \frac{\mathrm{d}\mathcal{L}(\phi(x))}{\mathrm{d}\phi}$ at a single sample x satisfies Axioms 1 and 2 iff it factors as*

$$\hat{J}_\eta = J_\phi S(x;\phi,\pi(\eta)) J_\eta^\pi, \tag{7}$$

where $S(x;\phi,\pi(\eta))$ is a $d \times K$ matrix for each x s.t.

$$\mathbb{E}_x[S(x)] = \Phi. \tag{8}$$

The proof is given in Appendix A. In particular, it is easy to see that for a linear loss $\mathcal{L}(x) = a^\mathsf{T}\phi(x)$, the true gradient is expressed as

$$\tfrac{d}{d\eta}\textstyle\sum_x a^\mathsf{T}\phi(x)p(x;\eta) = a^\mathsf{T}\Phi\frac{d\pi}{d\eta} = J_\phi\Phi J_\eta^\pi, \tag{9}$$

leading to the constraint (8). The theorem shows that estimators satisfying Axioms 1 and 2 are the ones that obey the chain rule and can be plugged in the backpropagation. It also shows that the class of GST estimators is extensive, parameterized by the set of matrices $\{S(x) \,|\, x \in \mathcal{X}\}$ satisfying (8). It encompasses many common estimators:

Proposition 1. *ST, FouST, DARN, and ZGR are GST estimators, with the respective S matrices:*

$$S^{\text{ST}}(x) = \Phi, \tag{10a}$$

$$S^{FouST}(x) = \Phi \,\mathrm{diag}(\tfrac{1}{Kp(x)}), \tag{10b}$$

$$S^{\text{DARN}(\tilde{\phi})}(x) = (\phi(x) - \tilde{\phi})\tfrac{1}{p(x)} e_x^\mathsf{T} + \tilde{\phi}\mathbf{1}^\mathsf{T}, \tag{10c}$$

$$S^{\text{ZGR}}(x) = \tfrac{1}{2}\left[S^{ST}(x) + S^{\text{DARN}(\bar{\phi}(\eta))}(x)\right]. \tag{10d}$$

Proposition 2. *ReinMax is a member of the GST family; furthermore, when the probability distribution is parameterized by a softmax function,* $S^{ReinMax}(x) = S^{ZGR}(x)$.

Proofs are provided in Appendix A. It remains to discuss the dependence of S on the embedding ϕ. Thus far, we have assumed that the embedding is given and fixed. The embedding may, however, be effectively modified by changing the loss function. Assume we have an estimator defined for an embedding ϕ_0. Let $\phi = T\phi_0 + b$ for some $T \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$. Then an estimator for the new embedding can be defined as $\hat{J}_\eta[\mathcal{L}, \phi, \pi, \omega] = \hat{J}_\eta[\mathcal{L} \circ (T\phi_0 + b), \phi_0, \pi, \omega]$. It is then expressed as

$$\hat{J}_\eta = J_\phi T S(x; \phi_0, \pi(\eta)) \tfrac{\mathrm{d}\pi}{\mathrm{d}\eta}. \tag{11}$$

The following proposition shows that this does not impose a restriction.

Proposition 3 (Projection)**.** *Any GST estimator for an embedding ϕ such that the embedding matrix Φ is full rank $d \times K$, $d < K$, can be represented as a GST estimator for the one-hot embedding ϕ_0* projected *via* (11) *with some T.*

Proof in Appendix A. While we have assumed that Φ is of full rank, if this is not the case, there remain redundant degrees of freedom in the estimator, which expand the class of GST estimators as defined. These will be addressed when identifying minimum variance estimators in the next section.

4 MINIMUM VARIANCE ESTIMATORS

We seek to derive minimum variance estimators subject to constraints on the bias. However, these quantities depend on the loss function and the underlying model $p(x; \eta)$, which both may be complex, *e.g.*, defined by deep networks. We therefore need to establish proxy criteria. Axiom 2 is already an example of a proxy for low bias: it requires zero bias for linear functions. Next, we propose proxy variance criteria, and in Sections 4.2 and 4.3 we consider their tradeoff vs. bias for quadratic loss functions.

4.1 Proxy Variance Criteria

Consider the variance of the i-th component of the gradient, $\mathbb{V}\left[\frac{\partial \mathcal{L}}{\partial \phi} S(x)\, J^\pi_{\eta_i}\right]$. It depends non-trivially on the derivative of the loss function with respect to the embedding. In order to simplify it and define a proxy metric, we consider linear loss functions $\mathcal{L}(\phi) = a^\mathsf{T}\phi$ and model a as a random variable, independent of x. As a *proxy variance criterion*, we then consider the expected variance summed over all gradient components:

$$\textstyle\sum_i \mathbb{E}_a \mathbb{V}\left[a^\mathsf{T} S(x)\, J^\pi_{\eta_i}\right]. \tag{12}$$

Note that the average variance is a meaningful criterion even if the average gradient over a is zero. By taking different distributions for a, different criteria can be designed. Expanding (12), it can be seen that the variance criterion is fully specified by the matrix $E = \mathbb{E}[aa^\mathsf{T}]$. As the most natural choice, we consider $a \sim \mathcal{N}(0, I)$, which corresponds to sampling all linear slopes isotropically and results in $E = I$. This leads to the *total variance* criterion:

$$\mathbb{V}^T[\hat{J}] = \mathbb{E}\|S(x) J^\pi_\eta\|^2_F, \tag{13}$$

which essentially sums the variances of all components of the Jacobian $S(x)J^\pi_\eta$ (see Appendix B.1 for details).

Finding the minimum variance estimator in the GST class without further constraints on the bias (beyond Axiom 2) has a trivial solution that does not depend on the choice of the proxy variance criterion:

Proposition 4. *ST* (2) *is the minimum variance GST estimator w.r.t. criterion* (12) *for any non-degenerate distribution of a.*

Proof. We have already shown that ST is a member of the GST class with the matrix $S(x) = \Phi$. Since it is deterministic, *i.e.*, it does not depend on x, its variance for a linear loss function is zero. □

Consequently, we seek optimal estimators under additional constraints on the bias.

4.2 Quadratically Unbiased Estimators

From the properties of ZGR (Shekhovtsov, 2023), we hypothesize that enforcing zero bias for all quadratic loss functions may be a beneficial constraint to impose.

Table 1: Overview of obtained solutions for hard and parametric quadratically unbiased MVEs.

Embedding	Proxy Variance	Complexity	Hard Solution	Parametric Solution
1D	Any E	$\mathcal{O}(K)$	Proposition C.3	Proposition C.1
One-hot	Total, $E = I$	$\mathcal{O}(K^3)$	Proposition C.4	Proposition C.2
One-hot	$E = V^\dagger$	$\mathcal{O}(K)$	ZGR, Appendix B.4	-
General	Total, $E = I$	$\mathcal{O}(d^2K)$	Appendix B.3	-

This is a natural extension of the linear unbiasedness constraint in Axiom 2 and serves as a meaningful proxy for low bias, as many loss functions are locally well-approximated by quadratics.

Under Axioms 1 and 2, zero bias for all quadratic loss functions can be ensured by satisfying the constraint for a spanning functional basis. One such basis is given by the set of monomials $\mathcal{Q} = \{\phi \mapsto \phi_i\phi_j \,|\, i, j \in 1 \dots d\}$. Any quadratic function can be expressed as a linear combination of these monomials and a linear function.

Given a proxy variance criterion $\bar{\mathbb{V}}[\hat{J}]$, we propose finding the corresponding Minimum Variance Estimator (MVE) in the GST class subject to quadratic unbiasedness constraints; i.e., to solve the following optimization problem:

$$\min_{\hat{J}} \mathbb{V}\big[\hat{J}\big] \quad \text{s.t.} \quad \mathbb{E}\big[\hat{J}\big] = \Phi, \quad B[\hat{J}] = 0, \qquad (14)$$

where $\hat{J}$ is parameterized by the tensor $\{S(x) \,|\, x \in \mathcal{X}\}$ as in (7), and $B[\hat{J}]$ is the tensor of biases of the estimator $\hat{J}$ w.r.t. the set $\mathcal{Q}$. Together with the linear unbiasedness constraint $\mathbb{E}\big[\hat{J}\big] = \Phi$, this ensures zero bias for all quadratic functions.

There are dK^2 variables (the elements of S) and d^2 linear constraints. A naive solution of the resulting system of equations would require $O((dK^2)^3)$ time, which is computationally prohibitive. Furthermore, it must be solved for a given distribution π, which is generally dependent on the model parameters and changes during optimization. However, by exploiting the special structure of the problem and specific cases of the embedding and proxy variance, much simpler solutions can be derived.

A particularly simple solution with $O(K)$ complexity is obtained by considering the proxy variance criterion defined by $E = V^\dagger$, where V is the covariance matrix of the embedding: $V = \mathbb{E}[\phi(x)\phi(x)^\top] - \bar{\phi}\bar{\phi}^\top$ and $V^\dagger$ is its pseudo-inverse.

Proposition 5. *The minimum variance w.r.t. proxy criterion* (12) *for $E = V^\dagger$ quadratically unbiased GST estimator for the one-hot embedding is ZGR.*

Proof is in Appendix B.4. This provides a novel insight: ZGR (and by extension ReinMax) can be derived from the unified principle of minimum variance quadratically unbiased estimation, albeit with a proxy criterion that depends dynamically on the covariance matrix of the embedding. The derivation in Appendix B.4 shows how the $V^\dagger$ criterion uniquely simplifies the equations, leading to a closed-form solution.

As a second special case, we consider 1D embeddings, *e.g.*, the integer embedding $\phi(x) \in \{0, \dots, K-1\}$. This serves as an example where the embedding dimension is smaller than the number of states. In such cases, we conjecture that a lower variance can be attained via a direct solution to (14) rather than by taking a generic solution for one-hot embedding and projecting it. For the 1D embedding, all variance criteria are equivalent as they differ at most by a scalar multiplier. The MVE solution (derived in Appendix C.1.1 as a special case of our more general approach) also has $O(K)$ complexity. Notably, this result differs from the projected ZGR estimator; specifically, the 1D MVE under the $V^\dagger$ criterion is distinct from the projected one-hot MVE under the same criterion. Empirically, this direct approach further reduces variance.

Finally, we can find the MVE for the one-hot embedding under the total variance criterion, which is a more natural choice than the one leading to ZGR, with a complexity of $O(K^3)$. This allows us to assess experimentally whether the $V^\dagger$ criterion of ZGR is inferior in practice. While computationally more demanding, it also naturally generalizes to the parametric analysis considered later in the paper. The complexity bottleneck is the inversion of a $K \times K$ matrix with a special structure, for which more efficient algorithms likely exist. For a concise summary, all the derived MVEs are summarized in Table 1.

4.3 Parametric Pareto-Optimal Family

We can obtain a family of estimators with different bias-variance trade-offs in a principled way by putting a bound on the quadratic bias while still minimizing the proxy variance:

$$\min_{\hat{J}} \mathbb{V}\big[\hat{J}\big] \quad \text{s.t.} \quad \mathbb{E}\big[\hat{J}\big] = \Phi, \quad \|B[\hat{J}]\|_F^2 \le b. \qquad (15)$$

By varying b, we can find minimum variance estimators at different trade-offs with the bias. This ef-

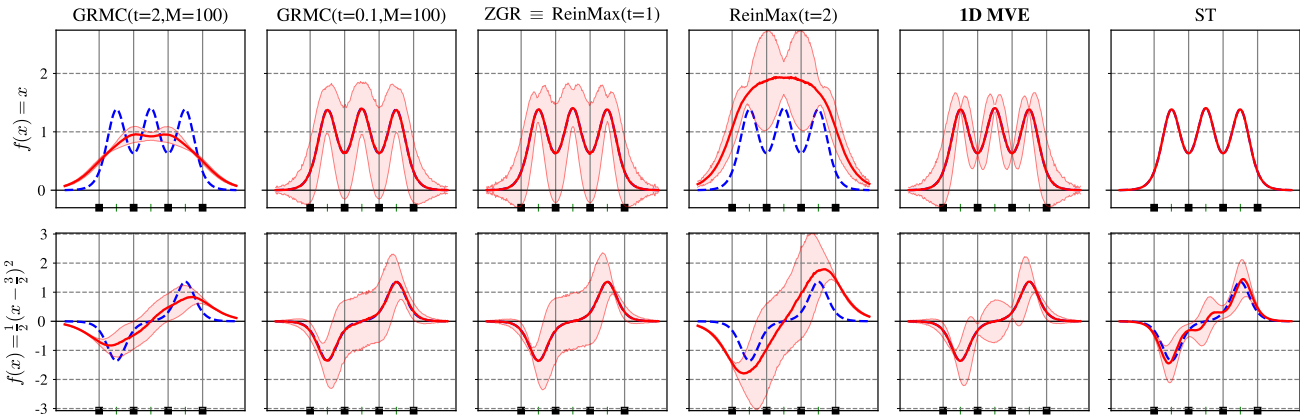


Figure 2: Bias and variance of different estimators for linear $\mathcal{L}(x) = x$ and quadratic $\mathcal{L}(x) = \frac{1}{2}(x - \frac{3}{2})^2$ loss functions of the integer embedding versus the input η to stochastic quantization. Black squares correspond to the set $\mathcal{X}$; the dashed blue line indicates the true gradient. The expected value of the estimator is shown as a solid red line and its standard deviation as a shaded area.

ficiently defines $\|B[\hat{J}]\|_F^2$ as our proxy bias measure summarizing the bias by one scalar. The family of optimal solutions to (15) for varying b coincides with the family of optimal solutions to the Lagrangian subproblem

$$\min_{\hat{J}} \left(\mathbb{V}\big[\hat{J}\big] + \tfrac{\tau}{2} \|B[\hat{J}]\|_F^2 \right) \quad \text{s.t.} \quad \mathbb{E}\big[\hat{J}\big] = \Phi \qquad (16)$$

for varying values of the Lagrange multiplier $\tau \geq 0$. We can therefore effectively find estimators with optimal trade-offs by solving (16) for different τ, although τ itself is not well interpretable. We call it *temperature* by analogy with the temperature parameter in GS, which also controls the bias-variance trade-off.

Solutions to (16) form a 1D manifold, which anneals from the ST estimator when $\tau = 0$ (as follows from Proposition 4) to the minimum variance quadratically unbiased estimator as $\tau \mapsto \infty$.

In Appendix C we obtain a general solution of complexity $O((dK)^3)$ and show that closed-form solutions can be derived and efficiently computed in the cases of 1D and one-hot embedding with the total variance criterion. These solutions have the same complexity as their hard quadratically unbiased counterparts. We have not found a simple solution for $V^\dagger$ variance that would extend ZGR to the parametric case.

Choosing τ in the parametric MVE(τ) allows selecting amongst Pareto-optimal estimators w.r.t. to proxy bias and variance criteria, as suitable for applications. The temperature τ can also be annealed: starting optimization with a small τ (low variance, higher bias) and progressively increasing it towards the end (decreasing the bias at the cost of higher variance).

5 EXPERIMENTS

To validate the proposed methods we conducted synthetic tests, in which it is possible to accurately compute the true gradient and measure bias and variance of different estimators. We then validate how the bias-variance tradeoff translates to utility in optimization in several test applications, evaluating 1D MVE, one-hot MVE and their parametric versions. Our implementation is publicly available[1].

5.1 Bias-Variance Tradeoff

1D MVE We evaluate bias and variance of 1D MVE as follows. The distribution $p(x;\eta)$ is defined by the relaxed quantization (Louizos et al., 2019), via $x = Q(\eta + Z)$, where $Z \sim$ Logistic with zero mean and std $1/3$ and Q rounds to the nearest integer in $\mathcal{X} = \{0, 1, \dots K-1\}$. For a given loss function $\mathcal{L}$ we plot bias and variance of different estimators versus η (o)Other estimators, defined for one-hot categorical variables, are applied via projection). The results are shown in Fig. 2. It verifies that 1D MVE is unbiased for quadratic loss functions and achieves a significantly lower variance than ZGR. At the same time, parametric GRMC(t) and especially ReinMax(t) are biased even for linear functions.

One-hot Parametric MVE Next we study the bias-variance tradeoff of different estimators for one-hot categorical variable x. We consider a random quadratic loss function $f(x) = \|L(x-c)^T\|_2^2$ where L is a random matrix and c is a random vector. We compare against ST, ZGR and GRMC estimator with

[1] https://github.com/James-Hooper123/Generalized-and-Optimal-Straight-Through-Estimators

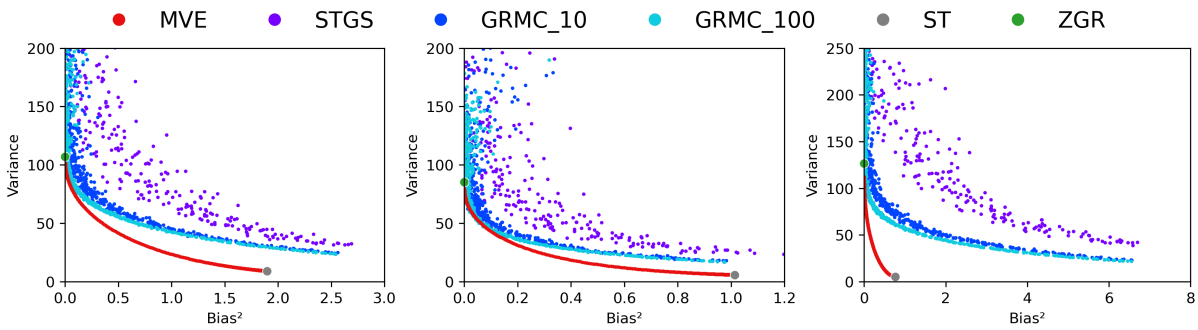


Figure 3: Bias-variance plots for three random quadratic loss functions of one-hot categorical variables with $K = 8$ states. The estimates of squared bias and variance for GRMC are themselves random, see text.

M inner Gumbel samples (for $M = 1$ it is identical to STGS). The latter is stochastic even for a fixed state x. We used 1000 trials to estimate its squared bias and variance in an unbiased way as detailed in Appendix D. Fig. 3 shows the bias-variance tradeoff for three random quadratic functions and $K = 8$ categorical states. For GRMC(t) and MVE(τ) we vary the respective temperature parameters, allowing us to obtain different bias-variance tradeoffs. We see that the parametric MVE(τ) family gives the Pareto-optimal frontier. While the variance of GRMC diverges in the limit $t \to 0$, MVE remains well-behaved for the whole temperature spectrum. The variance of the limiting quadratic unbiased MVE ($\tau = \infty$) is very similar to ZGR, suggesting that the exact proxy variance criterion does not matter much. In Appendix E we show also examples for $K = 4$ and $K = 16$ and observe that the gap widens with the number of categories.

It is important to acknowledge a fundamental limitation of the classical bias-variance evaluation in Fig. 3 concerning the gradient's scale. For instance, one could trivially construct an estimator that always outputs zero; while this yields zero variance and a bias equal to the true gradient, such a tradeoff is deceptive and obviously useless for optimization. Estimators that are biased for linear functions can freely scale the gradient magnitude up or down. This issue compromises the comparison against GRMC and ReinMax in high temperature regimes. We have not yet found a simple objective way to correct for this. Future work should identify the key tradeoff, more directly relevant for optimization.

5.2 Deep Quantized Networks

We have implemented the 1D MVE estimator in the framework for training quantized networks of Shekhovtsov and Obdrzalek (2026). We trained the TNet architecture proposed there for classification on ImageNet-100 (a 100-class subset of the ImageNet1K dataset). The network has 19 layers and activations of all layers are quantized to 3 bits (8 integer states) using the stochastic relaxation with logistic noise, as described above.

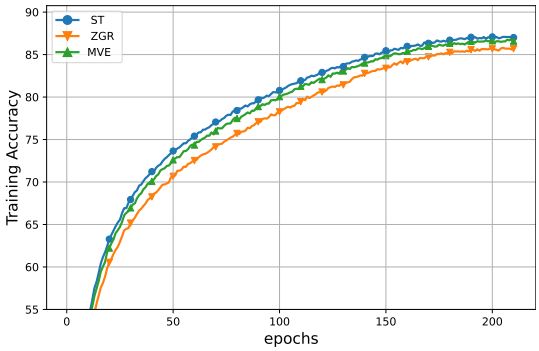


Figure 4: Evaluation of ST, ZGR, and MVE for training TNet with 3-bit activations on ImageNet-100. The plot shows the expected training accuracy (averaged over injected noise, mini-batches and data augmentation).

Fig. 4 shows the training performance in comparison with the ST and ZGR estimators. Consistent with the variance reduction observed in Fig. 2, we see that MVE improves over ZGR. This experiment corroborates that for training quantized networks a larger estimation bias is tolerable; consequently, ST achieves the highest performance, while the proposed MVE achieves nearly comparable results.

5.3 Polynomial Programming

We replicate the polynomial programming experiment of Liu et al. (2023), extending from binary to K-valued 1D embeddings. We note that their formulation decouples over variables and hence the problem can be presented using a single variable. The distribution of x is given by softmax(η), where $\eta \in \mathbb{R}^K$. The embed-

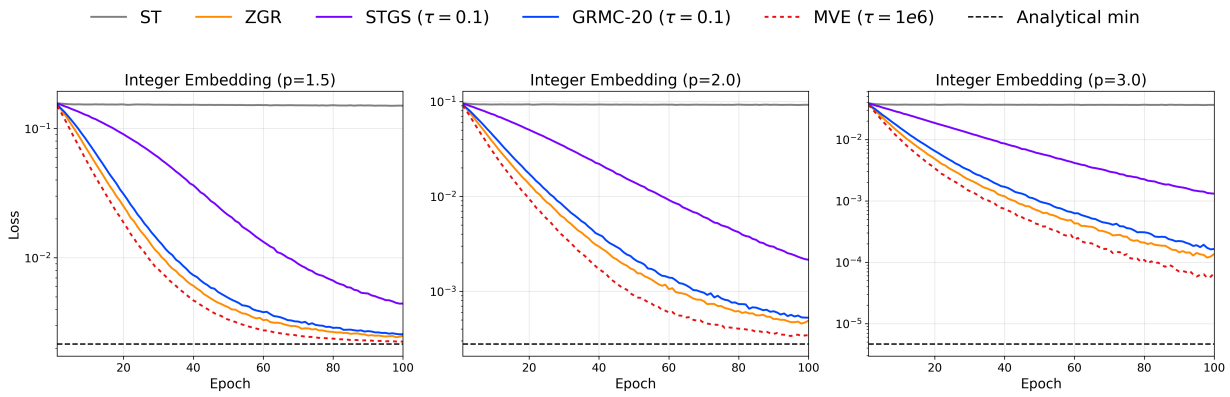


Figure 5: Polynomial programming, following Liu et al. (2023), for the 1D embedding with $K = 16$ points and powers $p = 1.5, 2.0, 3.0$. The plots show average NELBO over 128 independent runs.

ding ϕ maps x to $\{0, \frac{1}{K-1}, \ldots, 1\}$. The problem is to minimize $\mathbb{E}_x[|\phi(x) - c|^p]$ for $c = 0.45$ and some $p \geq 1$. Fig. 5 shows an experiment with initialization $\eta = 0$, Adam optimizer with $lr = 10^{-3}$. Each gradient is estimated using an average of 8 Monte Carlo samples. The results demonstrate that MVE in this case is more efficient than ReinMax while ST does not work. The latter is explained by a higher curvature and a more complex distribution, as compared to deep quantized networks above.

5.4 Variational Autoencoders

Next, we compare the performance of different gradient estimators on training a VAE (Kingma and Welling, 2014) with a discrete latent space $\mathcal{Z}^V$ of V categorical variables. The encoder and decoder architectures are MLPs with 2 hidden layers. The encoder $q_\phi(z|x)$ uses the softmax categorical model. Specifically, the encoder network outputs $K \times V$ logits. The decoder $p_\theta(x|z)$ uses either one-hot embeddings of z (the input to the decoder network is of size $K \times V$) or integer embeddings of z (in which case the input to the decoder network is of size V). The VAE is trained to maximize the evidence lower bound (ELBO):

$$\mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x) \| p(z)). \qquad (17)$$

Approximate gradient estimators are needed to estimate the gradient with respect to ϕ of the expectations over $q_\phi(z|x)$ using a single sample of z. We train it on the MNIST dataset using the AdamW optimizer with default weight decay of 10^{-2} (Loshchilov and Hutter, 2019), batch size 200, and a cosine learning rate schedule. See Appendix E for more details and discussion of running times in practice.

Fixed Schedules It is a common practice to apply STGS / GRMC with a temperature schedule so that in the beginning of the training the variance is reduced at the expense of the bias (high temperature), and towards the end of the training the bias is corrected (low temperature, low learning rate to tolerate variance). First, we compare the estimators using common exponential schedules of the form $\tau_n = \tau_0 \alpha^n$. GRMC schedule goes from 1 to 0.1 and MVE schedule goes from 1 to 10^6. Table 2 shows the results across different configurations of the latent space.

Our observations are as follows: ZGR (and by extension ReinMax) outperforms GRMC, consistently with prior work. In the categorical case, the proposed MVE Exp improves over ZGR for many categories and is tied with ReinMax(t). A detailed grid search over temperatures and learning rates in Fig. E.2 confirms the tie. In the integer case, MVE Exp significantly improves over ZGR and ReinMax(t) and the hard MVE is performing similarly, suggesting that the temperature schedule is not essential for this specific experimental setup.

Oracle Schedules To further test the potential of tuning the schedules for MVE and the baselines, we performed the following experiment (technical details in Appendix E.2.2). The experiment is conducted for VAE model with 8×64 one-hot latent variables. The optimal learning rate identified in the previous experiment is fixed (which is the same for MVE and GRMC: 10^{-3}). For each 25 epochs of training, we evaluate a range of temperatures and greedily select the temperature that achieves the best ELBO for this training period. We then proceed with the best result. We observe that while the oracle schedules result in similar performance for MVE, they yield a meaningful improvement for GRMC, suggesting that its original heuristic temperature schedule was not perfectly tuned. Despite this gain for the baseline, MVE still outperforms GRMC significantly (Fig. 6, right). The discovered schedules for both estimators demonstrate

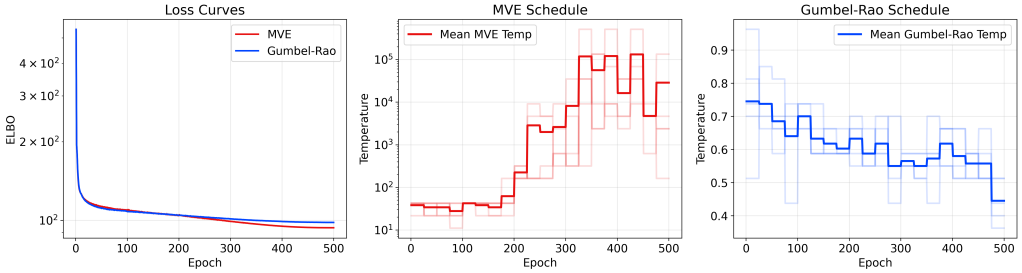
Table 2: VAE experiment with integer and one-hot categorical latent variables. The latent space size is varied such that the decoder capacity stays constant. Final training NELBO is averaged across 5 random seeds. Reported values represent the best result among 5 learning rates geometrically spaced from 10^{-2} to 10^{-4}. Superscripts indicate the optimal learning rate found: ^{a} learning rate $= 10^{-3}$; ^{b} learning rate $= 3.2 \times 10^{-3}$. The average standard deviation w.r.t. seeds across all configurations is ± 0.21; individual stds are included in Table E.1.

Integer Latents

Latents $V \times K$	128×4	128×16	128×64	128×256	128×1024
ST	112.9^b	113.7^b	113.8^b	113.8^b	113.8^b
GRMC-20 Exp	96.8^a	100.1^b	100.2^b	103.1^b	110.1^b
ZGR	96.0^b	94.5^a	94.8^a	95.7^a	97.9^a
ReinMax ($t = 1.2$)	96.3^b	94.8^a	95.0^a	96.0^a	98.2^b
ReinMax ($t = 1.4$)	97.0^b	95.3^a	95.7^a	96.9^a	99.7^b
1D MVE Exp	95.8^b	93.4^b	$\mathbf{93.2}^b$	$\mathbf{93.2}^b$	$\mathbf{93.2}^b$
1D MVE	$\mathbf{95.7}^b$	$\mathbf{93.1}^b$	93.3^b	93.3^b	93.3^b

One-hot Categorical Latents

Latents $V \times K$	128×4	32×16	8×64
ST	112.5^b	112.8^b	111.4^b
GRMC-20 Exp	96.3^a	95.8^b	101.5^b
ZGR	96.0^a	94.4^a	98.3^a
ReinMax ($t = 1.2$)	96.2^b	$\mathbf{93.5}^b$	93.7^b
ReinMax ($t = 1.4$)	96.6^b	93.9^a	93.7^b
MVE Exp	$\mathbf{95.4}^a$	94.6^b	$\mathbf{93.6}^b$



Average Final NELBO	
MVE	$\mathbf{93.6 \pm 0.3}$
GRMC-20	98.1 ± 0.4

Figure 6: Comparison of MVE(τ) and GRMC(t) under greedy oracle schedules for the categorical 8×64 latent space. The plots show the average training NELBO over 5 seeds and the found schedules for MVE and GRMC.

a consistent trend: transitioning from low-variance to low-bias. This provides empirical validation for the intuition behind manually designed temperature schedules.

6 CONCLUSION

We have proposed a systematic approach to designing approximate-chain-rule gradient estimators. Starting from the basic axioms has allowed us to describe a general class of straight-through-like estimators, which correspond to fixing the naive chain rule. We proposed the minimum variance principle for designing estimators and showed that ST and ZGR (equivalent to ReinMax) can be derived as optimal estimators under different proxy variance criteria subject to bias constraints. This provides a unified design principle and a new insight into these estimators. In contrast, ST was previously justified by linearization, ZGR was obtained from GRMC, and the ReinMax design was inspired by Heun's method.

We have also proposed a framework for designing estimators with a tunable bias-variance trade-off, which may be useful in practice. For every value of the temperature the estimator minimizes a proxy variance subject to bounded bias, and therefore achieves Pareto optimality according to the proxy criteria. This family matches or exceeds results in all conducted experiments: ST (MVE with $\tau = 0$) for quantization, and MVE with $\tau > 0$ for polynomial programming and discrete VAEs.

We have identified the problem structure and tractable solutions in several cases of interest. In particular, we have illustrated how to find minimum variance estimators specialized for a lower-dimensional embedding by studying the 1D case. We have verified that they can achieve lower variance in synthetic tests and that the reduced variance clearly translates to improved performance in optimization. Furthermore we have derived a solution for a general embedding for the total variance criterion with complexity $O(d^2K)$. This solution may be feasible in many applications where d is small. For the full-dimensional categorical embedding, the cost becomes $O(K^3)$, which could be prohibitive for large K. Nevertheless, experiments suggest that the improvement over ZGR becomes more substantial for larger K. This motivates investigating potential gains in other applications and, in cases where such improvements are observed, looking for more practical approximate solutions.

Acknowledgements

We sincerely acknowledge the support by Toyota Motor Europe and Czech Science Foundation Grant number GA24-12697S.

References

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

Matthieu Courbariaux and Yoshua Bengio. BinaryNet: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830, 2016.

Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. In *ICML*, 2014.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *ICLR*, 2021.

Koen Helwegen, James Widdicombe, Lukas Geiger, Zechun Liu, Kwang-Ting Cheng, and Roeland Nusselder. Latent weights do not exist: Rethinking binarized neural network optimization. In *NeurIPS*, 2019.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *J. Mach. Learn. Res.*, 18(1), 2017.

Iris A.M. Huijben, Bastiaan S. Veeling, and Ruud J.G. van Sloun. Deep probabilistic subsampling for task-adaptive compressed sensing. In *ICLR*, 2020.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2017.

Siddhant M. Jayakumar, Razvan Pascanu, Jack W. Rae, Simon Osindero, and Erich Elsen. Top-KAST: top-k always sparse training. In *NeurIPS*, 2020.

Ahmed Khalil, Robert Piechocki, and Raul Santos-Rodriguez. LL-VQ-VAE: Learnable lattice vector-quantization for efficient representations. *ArXiv*, 2023.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *NeurIPS*. 2017.

Liyuan Liu, Chengyu Dong, Xiaodong Liu, Bin Yu, and Jianfeng Gao. Bridging discrete and backpropagation: Straight-through and beyond. In *NeurIPS*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, and Max Welling. Relaxed quantization for discretized neural networks. In *ICLR*, 2019.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.

Max B Paulus, Chris J. Maddison, and Andreas Krause. Rao-blackwellizing the straight-through Gumbel-softmax gradient estimator. In *ICLR*, 2021.

Adeel Pervez, Taco Cohen, and Efstratios Gavves. Low bias low variance gradient estimates for boolean stochastic networks. In *ICML*, 2020.

Semyon Savkin, Eitan Porat, Or Ordentlich, and Yury Polyanskiy. Nestquant: nested lattice quantization for matrix products and LLMs. In *ICML*, 2025.

Alexander Shekhovtsov. Cold analysis of Rao-Blackwellized straight-through Gumbel-softmax gradient estimator. In *ICML*, 2023.

Alexander Shekhovtsov and Stepan Obdrzalek. Scalable binary-quantized neural networks for energy-efficient vision, 2026. To appear.

Alexander Shekhovtsov and Viktor Yanush. Reintroducing straight-through estimators as principled methods for stochastic binary networks. In *GCPR*, 2021.

Seiya Tokui and Issei Sato. Evaluating the variance of likelihood-ratio gradient estimators. In *ICML*, 2017.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification for your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such a case we will ask you to upload it during the author response period and include it in camera-ready (if accepted).

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Source code, with specification of all dependencies, including external libraries. [No]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A PROOFS: GENERALIZED ST THEOREM

Let us recall the setup. Random variable x takes values in a discrete set $\mathcal{X}$ with $|\mathcal{X}| = K$. Distribution $p(x;\eta)$ is parameterized by η and the corresponding probabilities form a vector $\pi(\eta) \in \Delta^{K-1}$.

Since π is in the probability simplex, the full derivative with respect to π is not needed, only its projection to the simplex is relevant. More precisely this is captured by the following lemma.

Lemma A.1. *Let Δ^{K-1} denote the probability simplex $\{\pi \in \mathbb{R}^K \,|\, \pi \geq 0,\ \sum_k \pi_k = 1\}$. Let $\pi(\eta) \in \Delta^{K-1}$ for any η and differentiable at η_0. Let $J = \frac{\partial \pi(\eta_0)}{\partial \eta}$. Then $\mathbf{1}^\mathsf{T} J = \mathbf{0}$.*

Proof. For a differentiable mapping we have that $\pi(\eta_0+h) = \pi(\eta_0)+Jh+\boldsymbol{o}(\|h\|)$. Since both $\pi(\eta_0)$ and $\pi(\eta_0+h)$ are in Δ^{K-1}, it must be $\mathbf{1}^\mathsf{T}(\pi(\eta_0+h)-\pi(\eta_0)) = 0$. It follows that $\mathbf{1}^\mathsf{T} Jh = 1^\mathsf{T}\boldsymbol{o}(\|h\|) = o(\|h\|)\ \forall h$. This can only hold when $\mathbf{1}^\mathsf{T} J = \mathbf{0}$. □

In other words, if the derivative in π is correct up to a vector $b\mathbf{1}^\mathsf{T}$ for some $b \in \mathbb{R}$, it is equally good for estimating the derivative in η. We will therefore consider the derivative in π as an intermediate result, which may be manipulated up to such constant additive to all coordinates to our convenience.

Theorem 1 (Generalized ST)**.** *An estimator in the sense of Definition 1 that depends on $\mathcal{L}$ only through the derivative $J_\phi = \frac{\mathrm{d}\mathcal{L}(\phi(x))}{\mathrm{d}\phi}$ at a single sample x satisfies Axioms 1 and 2 iff it factors as*

$$\hat{J}_\eta = J_\phi S(x;\phi,\pi(\eta)) J_\eta^\pi, \tag{7}$$

where $S(x;\phi,\pi(\eta))$ is a $d \times K$ matrix for each x s.t.

$$\mathbb{E}_x[S(x)] = \Phi. \tag{8}$$

Proof. Let us show that any estimator based on the derivative $J_\phi = \frac{\mathrm{d}\mathcal{L}(\phi(x))}{\mathrm{d}\phi}$ at a single sample x satisfies Axioms 1 and 2 can be represented in the form (7)-(8).

Dependence on J_ϕ. If the estimator is based on the derivative J_ϕ, the linearity w.r.t. loss implies that it must be linear in this derivative, *i.e.* take the form

$$\hat{J}_\pi = J_\phi \tilde{S}(x;\phi,\pi), \tag{18}$$

where $\tilde{S}$ is a matrix of the size $d \times K$ depending on the value of x, function ϕ and vector of probabilites $\pi \in \Delta^K$.

Expectation of $\tilde{S}$. Let $\mathcal{L}(\phi(x)) = a^\mathsf{T}\phi(x)$ be a linear loss function. From Axiom 2 we obtain the following. The true derivative expresses as follows

$$J_{\pi_j} := \tfrac{\mathrm{d}}{\mathrm{d}\pi_j} \textstyle\sum_k \pi_k \mathcal{L}(k) = \sum_k [\![j{=}k]\!] a^\mathsf{T}\phi(k) = a^\mathsf{T}\phi(j); \tag{19}$$

$$J_\eta = J_\pi J_\eta^\pi = a^\mathsf{T}\Phi J_\eta^\pi. \tag{20}$$

The expectation of the estimator for $J_\phi = a$ expresses as

$$\mathbb{E}_x[\hat{J}_\eta] = \mathbb{E}_x[J_\phi \tilde{S}(x;\phi,\pi)] J_\eta^\pi = a^\mathsf{T}\mathbb{E}_x[\tilde{S}(x;\phi,\pi)] J_\eta^\pi. \tag{21}$$

We must assume that a can be any loss vector, therefore there holds

$$\Phi J_\eta^\pi = \mathbb{E}_x[\tilde{S}(x;\phi,\pi)] J_\eta^\pi. \tag{22}$$

Assuming that the model can represent any distribution $\pi \in \Delta^{K-1}$, J_η^π must have rank $K-1$. Then according to Lemma A.1, its null space is spanned by $\mathbf{1}$. Therefore there must exist a vector $b(\pi) \in \mathbb{R}^K$ such that

$$\mathbb{E}_x[\tilde{S}(x)] = \Phi + b(\pi)\mathbf{1}^\mathsf{T}. \tag{23}$$

We then let $S(x;\phi,\pi) = \tilde{S}(x;\phi,\pi) - b(\pi)\mathbf{1}^\mathsf{T}$. It satisfies (7) and (8).

We have shown the "only if" part of the theorem. The "if" part is straightforward by checking that any estimator of the form (7)-(8) satisfies Axioms 1 and 2. □

Proposition 1. *ST, FouST, DARN, and ZGR are GST estimators, with the respective S matrices:*

$$S^{\mathrm{ST}}(x) = \Phi, \tag{10a}$$
$$S^{FouST}(x) = \Phi \operatorname{diag}(\tfrac{1}{Kp(x)}), \tag{10b}$$
$$S^{\mathrm{DARN}(\tilde\phi)}(x) = (\phi(x) - \tilde\phi)\tfrac{1}{p(x)} e_x^\mathsf{T} + \tilde\phi \mathbf{1}^\mathsf{T}, \tag{10c}$$
$$S^{\mathrm{ZGR}}(x) = \tfrac{1}{2}\left[S^{ST}(x) + S^{\mathrm{DARN}(\bar\phi(\eta))}(x)\right]. \tag{10d}$$

Proof. **ST**. Let us verify ST can be defined by $S(x) = \Phi$. In this case the form (7) can be written as

$$\hat J_\eta^{\mathrm{ST}} = J_\phi \Phi J_\eta^\pi = J_\phi \textstyle\sum_k \phi(k) \frac{d\pi_k}{d\eta} = J_\phi \frac{d}{d\eta} \sum_k \phi(k)\pi_k = J_\phi \frac{d\bar\phi(\eta)}{d\eta}, \tag{24}$$

which matches the definition of ST (2). Clearly, it satisfies (8) since $\mathbb{E}_x[S(x)] = \Phi$.

FouST. FouST Pervez et al. (2020) can be defined by $S(x) = \Phi \operatorname{diag}(\frac{1}{Kp(x)})$. Indeed, in this case the form (7) can be written as

$$\hat J_\eta^{\mathrm{FouST}} = J_\phi S(x) \tfrac{d\pi}{d\eta} = J_\phi \Phi \operatorname{diag}(\tfrac{1}{Kp(x)}) \tfrac{d\pi}{d\eta} = \tfrac{1}{Kp(x)} J_\phi \tfrac{d\bar\phi(\eta)}{d\eta}. \tag{25}$$

Its expected value is

$$\mathbb{E}_x[S(x)] = \textstyle\sum_x p(x) \Phi \operatorname{diag}(\frac{1}{Kp(x)}) = \Phi \operatorname{diag}(\frac{1}{K}\sum_x 1) = \Phi. \tag{26}$$

DARN. DARN Gregor et al. (2014) in a general form for an arbitrary $\tilde\phi \in \mathbb{R}^k$ is defined as

$$\hat J_\eta^{\mathrm{DARN}(\tilde\phi)} = J_\phi (\phi(x) - \tilde\phi) \tfrac{d \log p(x;\eta)}{d\eta}. \tag{27}$$

Its S matrix can be identified as

$$S(x) = (\phi(x) - \tilde\phi)\tfrac{1}{p(x)} e_x^\mathsf{T} + \tilde\phi \mathbf{1}^\mathsf{T}, \tag{28}$$

where e_x is the standard basis vector for class x and $\tilde\phi \mathbf{1}^\mathsf{T}$ is a shift correction, which cancels when multiplied with $\frac{d\pi}{d\eta}$ (see proof of Theorem 1). Let's verify that it satisfies (8):

$$\mathbb{E}_x[S(x)] = \textstyle\sum_x p(x) \Big((\phi(x) - \tilde\phi)\frac{1}{p(x)} e_x^\mathsf{T} + \tilde\phi \mathbf{1}^\mathsf{T} \Big) = \sum_x (\phi(x) - \tilde\phi) e_x^\mathsf{T} + \tilde\phi \mathbf{1}^\mathsf{T} = \Phi. \tag{29}$$

ZGR. ZGR (Shekhovtsov, 2023) is defined as

$$\hat J_\eta^{\mathrm{ZGR}} = \tfrac{1}{2}\Big(\hat J_\eta^{\mathrm{ST}} + \hat J_\eta^{\mathrm{DARN}}(\bar\phi(\eta)) \Big). \tag{30}$$

It's S matrix is respectively,

$$S(x) = \tfrac{1}{2}\Big(S^{\mathrm{ST}}(x) + S^{\mathrm{DARN}(\bar phi)}(x) \Big). \tag{31}$$

It is a member of GST since both ST and DARN are. □

Proposition 2. *ReinMax is a member of the GST family; furthermore, when the probability distribution is parameterized by a softmax function, $S^{ReinMax}(x) = S^{ZGR}(x)$.*

Proof. We begin by recalling the definition of Reinmax:

$$\nabla_{\mathrm{Reinmax}} = 2 \cdot \hat\nabla^{\frac{\pi+D}{2}} - \tfrac{1}{2}\hat\nabla_{\mathrm{ST}} \tag{32}$$
$$\hat\nabla^{\frac{\pi+D}{2}} = \tfrac{\delta f(D)}{\delta D} \cdot \big((\pi_D \cdot \mathbf{1}^\top) \odot I - \pi_D \pi_D^\top \big) \tag{33}$$
$$\pi_D = \tfrac{\pi+D}{2}, \quad D = e_x, \quad x \sim \mathrm{Cat}(\pi), \quad \pi = \mathrm{softmax}(\theta) \tag{34}$$

Rewriting in our notation

$$\hat{\nabla}^{\frac{\pi+D}{2}} = \frac{\partial f(D)}{\partial D} \cdot \left(\operatorname{diag}(\pi_D) - \pi_D \pi_D^\top\right) \tag{35}$$
$$= \frac{\partial L(\phi(x))}{\partial \phi} \cdot \left[\tfrac{1}{2}\left(\operatorname{diag}(\pi) + \operatorname{diag}(e_x)\right) - \tfrac{1}{4}\left(\pi\pi^\top + e_x e_x^\top + \pi e_x^\top + e_x \pi^\top\right)\right] \tag{36}$$

We proceed with some straightforward manipulation

$$\hat{\nabla}^{\frac{\pi+D}{2}} = \tfrac{1}{2}\frac{\partial L(\phi(x))}{\partial \phi} \cdot \left(\operatorname{diag}(\pi) - \pi\pi^\top\right) + \tfrac{1}{4}\frac{\partial L(\phi(x))}{\partial \phi} \cdot \left(\pi\pi^\top + 2\operatorname{diag}(e_x) - e_x e_x^\top - \pi e_x^\top - e_x \pi^\top\right) \tag{37}$$
$$= \tfrac{1}{2}\frac{\partial L(\phi(x))}{\partial \phi} \cdot \left(\operatorname{diag}(\pi) - \pi\pi^\top\right) + \tfrac{1}{4}\frac{\partial L(\phi(x))}{\partial \phi} \cdot \left(\pi\pi^\top + e_x e_x^\top - \pi e_x^\top - e_x \pi^\top\right) \tag{38}$$
$$= \tfrac{1}{2}\hat{\nabla}_{\mathrm{ST}} + \tfrac{1}{4}\frac{\partial L(\phi(x))}{\partial \phi} \cdot \left(\pi\pi^\top + e_x e_x^\top - \pi e_x^\top - e_x \pi^\top\right) \tag{39}$$

Substituting this expression into the expression for Reinmax gives

$$\nabla_{\mathrm{Reinmax}} = 2 \cdot \hat{\nabla}^{\frac{\pi+D}{2}} - \tfrac{1}{2}\hat{\nabla}_{\mathrm{ST}} \tag{40}$$
$$= \tfrac{1}{2}\hat{\nabla}_{\mathrm{ST}} + \tfrac{1}{2}\frac{\partial L(\phi(x))}{\partial \phi} \cdot \left(\pi\pi^\top + e_x e_x^\top - \pi e_x^\top - e_x \pi^\top\right) \tag{41}$$
$$= \tfrac{1}{2}\hat{\nabla}_{\mathrm{ST}} + \tfrac{1}{2}\frac{\partial L(\phi(x))}{\partial \phi} \cdot \left(\tfrac{1}{p(e_x)}(e_x - \pi)e_x^\top\right)\left(\operatorname{diag}(\pi) - \pi\pi^\top\right) \tag{42}$$

Therefore in the case where our probability distribution is parameterized by the softmax function $S_{\mathrm{Reinmax}}(x) = \frac{1}{2}\left[I + \frac{1}{p(x)}(e_x - \pi)e_x^\top\right] = S_{ZGR}(x)$ □

Proposition 3 (Projection)**.** *Any GST estimator for an embedding ϕ such that the embedding matrix Φ is full rank $d \times K$, $d < K$, can be represented as a GST estimator for the one-hot embedding ϕ_0* projected *via* (11) *with some T.*

Proof. Let Φ be the embedding matrix for ϕ. The embedding matrix for ϕ_0 is $\Phi_0 = I$. Let $S(x)$ be the matrix defining the GST estimator for ϕ. It satisfies $\mathbb{E}_x[S(x)] = \Phi$. Define

$$S_0(x) = \Phi^\dagger(S(x) - \Phi) + I, \tag{43}$$

where $\Phi^\dagger$ is the Moore-Penrose pseudoinverse of Φ. Since Φ is full rank $d \times K$, it is the right-inverse, *i.e.* $\Phi\Phi^\dagger = I$. Therefore S is indeed a projection of S_0:

$$\Phi S_0(x) = \Phi\Phi^\dagger(S(x) - \Phi) + \Phi = S(x). \tag{44}$$

Further, let us verify that S_0 satisfies the constraint (8) for $\Phi_0 = I$:

$$\mathbb{E}_x[S_0(x)] = \Phi^\dagger(\mathbb{E}_x[S(x)] - \Phi) + I \tag{45a}$$
$$= \Phi^\dagger(\Phi - \Phi) + I = I. \tag{45b}$$

Therefore S_0 defines a GST estimator for the one-hot embedding ϕ_0 and S is its projection. □

B MINIMUM VARIANCE GST ESTIMATORS UNBIASED FOR QUADRATIC FUNCTIONS

B.1 Proxy Variance

Consider the variance of i'th component of the gradient, $\mathbb{V}\left[\frac{\partial \mathcal{L}}{\partial \phi}\, S(x)\, J^\pi_{\eta_i}\right]$. It depends non-trivially on the derivative of the loss function with respect to the embedding. In order to simplify it and define a proxy metric, we

consider linear loss functions $\mathcal{L}(\phi) = a\top\phi$ and model a as a random variable, independent of x, denoted as $a \sim \mathcal{D}$. We then consider the expected variance criterion, summed over all gradient coordinates:

$$\textstyle\sum_i \mathbb{E}_{a\sim\mathcal{D}} \mathbb{V}\Big[a\top S(x)\, J^\pi_{\eta_i}\Big]. \tag{46}$$

Using the variance decomposition through second moment, and knowing that for GST estimators $\mathbb{E}[\hat{J}] = g$ — the true gradient, we can rewrite this as

$$\textstyle\sum_i \mathbb{E}_{a\sim\mathcal{D},x\sim p}\Big[\Big(a\top S(x)\, J^\pi_{\eta_i}\Big)^2 - g_i^2\Big]. \tag{47}$$

Since g is the true gradient, which does not depend on $S(x)$, it can be ignored when considering optimization of the proxy variance in $\hat{J}$, and the criterion can be simplified to

$$\textstyle\sum_i \mathbb{E}_x\Big[J^\pi_{\eta_i} S(x)\top \mathbb{E}[aa^\mathsf{T}] S(x) J^\pi_{\eta_i}\Big]. \tag{48}$$

This criterion is appealing because it decouples the variance from the actual loss function, and depends only on the matrix $E = \mathbb{E}[aa^\mathsf{T}]$ associated with the distribution of likely linear approximations. For the isotropic $a \sim \mathcal{N}(0, I)$ there holds $\mathbb{E}[aa^\mathsf{T}] = I$ and we obtain

$$\textstyle\sum_i \mathbb{E}_x\Big[J^\pi_{\eta_i} S(x)\top S(x) J^\pi_{\eta_i}\Big] = \mathbb{E}_x\Big[\|S(x) J^\pi_\eta\|_F^2\Big]. \tag{49}$$

B.2 General Setup and the Representation Theorem

We formulate the Lagrangian for a general embedding and a general distribution over the gradient a. Our objective is to minimize the variance of the estimator (48), subject to the following constraints:

1. A constraint ensuring unbiasedness for linear functions
2. A basis of constraints ensuring unbiasedness for purely quadratic functions

We now derive our two constraints by equating the expected value of the estimator to the true derivative of the expected loss for linear and quadratic functions, respectively:

$$\forall a \in \mathbb{R}^d \qquad \mathbb{E}[\hat{J}_\eta] = \textstyle\sum_x p(x) a^\top S(x) J^\pi_\eta \overset{!}{=} \frac{d}{d\eta} \sum_x p(x) a^\top \phi(x), \tag{50a}$$

$$\forall i,j \in [d] \qquad \mathbb{E}[\hat{J}_\eta] = \textstyle\sum_x p(x) \phi(x)^\top (Q + Q^\top) S(x) J^\pi_\eta \overset{!}{=} \frac{d}{d\eta} \sum_x p(x) \phi(x)^\top Q \phi(x), \tag{50b}$$

where Q is the matrix having a 1 at the (i,j)-th entry and zeros elsewhere. Decomposing the constraints for each component of η, we obtain:

$$\forall j,k \quad \textstyle\sum_x p(x) R(x)_{jk} = A_{jk}, \tag{51a}$$

$$\forall i,j,k \quad \textstyle\sum_x p(x) \left[\phi(x)_i (R(x))_{jk} + \phi(x)_j (R(x))_{ik}\right] = B_{ijk}, \tag{51b}$$

where

$$A_{jk} = \textstyle\frac{d}{d\eta_k} \sum_x p(x) \phi(x)_j, \quad B_{ijk} = \frac{d}{d\eta_k} \sum_x p(x) \phi(x)_i \phi(x)_j. \tag{52}$$

The Lagrangian for the constrained problem is given by:

$$\begin{aligned} L(R, \alpha, \beta) =& \mathbb{E}_{a,x} \|a^\top R(x)\|^2 \\ &- \textstyle\sum_{j,k} \alpha_{jk} \left(A_{jk} - \sum_x p(x) R(x)_{jk}\right) \\ &- \textstyle\sum_{i,j,k} \beta_{ijk} \left(B_{ijk} - \sum_x p(x) \left[\phi(x)_i R(x)_{jk} + \phi(x)_j R(x)_{ik}\right]\right) \end{aligned} \tag{53}$$

We note that the quadratic unbiasedness constraint (50b) is symmetric in i, j; hence, we may assume without loss of generality that the Lagrange multipliers β are symmetric. Furthermore, in the case where $E = \mathbb{E}_{a\sim D}[aa^\top]$

has full rank we observe that our optimization problem is equivalent to minimizing the norm of R in a Hilbert space equipped with the inner product

$$\langle f, g\rangle_{\mathcal{H}} = \mathbb{E}_x \left[\mathrm{Tr}\left(f(x)^\top E g(x)\right)\right], \tag{54}$$

subject to linear constraints. In the case where E is rank deficient we may decompose the constraints with respect to the null and row space of E. The objective lies entirely within the row space and we can write our Lagrangian in terms of a similar inner product. The null space constraints can be solved separately as they do not affect the objective. Consequently, we invoke the classical Representer Theorem (Kimeldorf & Wahba, 1971; Schölkopf et al., 2001), specialized to the setting of linear equality constraints.

Theorem B.1 (Representer Theorem for Penalized Constraints)**.** *Let $\mathcal{H}$ be a Hilbert space of functions R. Let $\{L_m\}_{m=1}^M$ be a set of M bounded linear functionals, let $\mathbf{C} = [C_1, \ldots, C_M]^\top$ be the vector of constraint values, and let $\mu_m > 0$ for $m = 1, \ldots, M$ be penalty weights. Consider the optimization problem:*

$$R^* = \arg\min_{R\in\mathcal{H}} \ \|R\|_{\mathcal{H}}^2 + \sum_{m=1}^{M} \mu_m \big(L_m(R) - C_m\big)^2$$

The unique optimal solution R^* is a linear function of the constraint values $\mathbf{C}$. In the limit $\mu_m \to \infty$, this result extends to the case of hard constraints.

Proof (Proof of Theorem B.1). By the Riesz representation theorem, each bounded linear functional L_m has a unique representer $\ell_m \in \mathcal{H}$ such that $L_m(R) = \langle R, \ell_m\rangle_{\mathcal{H}}$. The objective becomes:

$$\mathcal{F}(R) = \|R\|_{\mathcal{H}}^2 + \textstyle\sum_{m=1}^{M} \mu_m \left(\langle R, \ell_m\rangle_{\mathcal{H}} - C_m\right)^2.$$

Taking the functional derivative with respect to R and setting it to zero yields:

$$2R + 2\textstyle\sum_{m=1}^{M} \mu_m \left(\langle R, \ell_m\rangle_{\mathcal{H}} - C_m\right) \ell_m = 0.$$

Thus, the optimal solution has the form $R^* = \sum_{m=1}^{M} \alpha_m \ell_m$. Substituting this into the optimality condition:

$$\alpha_m = \mu_m \left(C_m - \textstyle\sum_{k=1}^{M} \alpha_k \langle \ell_k, \ell_m\rangle_{\mathcal{H}}\right).$$

This can be written as the linear system $(D_\mu^{-1} + G)\boldsymbol{\alpha} = \mathbf{C}$, where G is the Gram matrix with entries $G_{km} = \langle \ell_k, \ell_m\rangle_{\mathcal{H}}$ and $D_\mu = \mathrm{diag}(\mu_1, \ldots, \mu_M)$. Since G is positive semi-definite and D_μ^{-1} is strictly positive definite, their sum $(D_\mu^{-1} + G)$ is strictly positive definite and hence invertible, giving $\boldsymbol{\alpha} = (D_\mu^{-1} + G)^{-1}\mathbf{C}$, which is a linear function of $\mathbf{C}$. Therefore, $R^* = \sum_{m=1}^{M} \alpha_m \ell_m$ is a linear function of the constraint values $\mathbf{C}$.

□

Therefore our solution is linear in the constraints and therefore linear in J_η^π. As a result, we can determine the linear operator S by setting $J_\eta^\pi = I$ and solving for the matrix S. Furthermore, we observe that the Lagrangian is separable in k and therefore it suffices to consider some fixed k. Our new simplified Lagrangian is as follows:

$$\begin{aligned} L(S, \alpha, \beta) = \mathbb{E}_{a\sim D, x}&\left[\left(a^\top S(x) e_k\right)^2\right] \\ &- \textstyle\sum_j \alpha_j \left(\Phi_{jk} - \sum_x p(x) S(x)_{jk}\right) \\ &- \textstyle\sum_{ij} \beta_{ij} \left(\Phi_{ik}\Phi_{jk} - \sum_x p(x) \left[\phi(x)_i S(x)_{jk} + \phi(x)_j S(x)_{ik}\right]\right) \end{aligned} \tag{55}$$

Note that the constraints and solution are restricted to the row space of E, though we omit this indication to simplify notation. We now take the derivative of the Lagrangian:

$$\tfrac{\partial L}{\partial S(x)_{jk}} = 2p(x)\mathbb{E}_{a\sim D}\left[a_j \left(a^\top S(x) e_k\right)\right] - \alpha_j p(x) - p(x) \textstyle\sum_i \left(\beta_{ij} + \beta_{ji}\right) \phi(x)_i \tag{56}$$

Writing this in vector format yields

$$\frac{\partial L}{\partial S(x)_{:k}} = 2p(x)\mathbb{E}_{a\sim D}\left[aa^\top S(x)e_k\right] - \alpha p(x) - p(x)\sum_i \left(\beta_{i:} + \beta_{:i}\right)\phi(x)_i \tag{57}$$

Setting this to zero and rearranging gives

$$S(x)_{:k} = E^\dagger\left(\tfrac{1}{2}\alpha + \sum_i \tfrac{1}{2}(\beta_{i:} + \beta_{:i})\phi(x)_i\right) \tag{58}$$

Assuming symmetry in β without loss of generality, we rewrite the expression in matrix notation as

$$S(x)_{:k} = E^\dagger\left(\tfrac{1}{2}\alpha + \beta\Phi e_x\right) \tag{59}$$

Substituting this into the first constraint implies

$$\sum_x p(x)E^\dagger\left(\tfrac{1}{2}\alpha + \beta\Phi e_x\right) = \Phi_{:k} \tag{60}$$

$$E^\dagger\left(\tfrac{1}{2}\alpha + \sum_i \beta\Phi p\right) = \Phi_{:k} \tag{61}$$

$$\tfrac{1}{2}E^\dagger\alpha = \Phi_{:k} - E^\dagger\beta\Phi p \tag{62}$$

Substituting back and manipulating terms results in

$$S(x)_{:k} = \Phi_{:k} + E^\dagger\beta\Phi(e_x - p) \tag{63}$$

Inserting this into the second constraint yields

$$\sum_x p(x)\left[\phi(x)_i\left(\Phi_{jk} + E^\dagger_{j:}\beta\Phi(e_x - p)\right) + \phi(x)_j\left(\Phi_{ik} + E^\dagger_{i:}\beta\Phi(e_x - p)\right)\right] = \Phi_{ik}\Phi_{jk} \tag{64}$$

Re-indexing the terms leads to

$$\sum_x p_x\left[\Phi_{ix}\left(\Phi_{jk} + E^\dagger_{j:}\sum_z \beta_{:z}(\Phi_{zx} - M_z)\right) + \Phi_{jx}\left(\Phi_{ik} + E^\dagger_{i:}\sum_z \beta_{:z}(\Phi_{zx} - M_z)\right)\right] = \Phi_{ik}\Phi_{jk} \tag{65}$$

Expanding and evaluating the sum over x gives

$$M_i\Phi_{jk} + E^\dagger_{j:}\sum_z \beta_{:z}(M_{iz} - M_iM_z) + M_j\Phi_{ik} + E^\dagger_{i:}\sum_z \beta_{:z}(M_{jz} - M_jM_z) = \Phi_{ik}\Phi_{jk} \tag{66}$$

Rearranging and substituting the definition of the embedding covariance matrix V yields

$$E^\dagger_{j:}\sum_z \beta_{:z}V_{zi} + E^\dagger_{i:}\sum_z \beta_{:z}V_{zj} = \frac{\partial V_{ij}}{\partial p_k} \tag{67}$$

Finally, rewriting in matrix notation reveals the symmetric structure

$$(E^\dagger\beta V)_{ij} + (E^\dagger_{i:}\beta V)_{ji} = \frac{\partial V_{ij}}{\partial p_k} \tag{68}$$

$$E^\dagger\beta V + (E^\dagger\beta V)^\top = \frac{\partial V}{\partial p_k} \tag{69}$$

$$V\beta E^\dagger + E^\dagger\beta^\top V = \frac{\partial V}{\partial p_k}. \tag{70}$$

This is a type of generalized Lyapunov equation. Since such equations generally do not admit explicit matrix-algebraic solutions, we proceed on a case-by-case basis, utilizing the properties of E to simplify the solution. We first note that our solutions for the design matrix are of the following form:

$$S(x)_{:k} = \Phi_{:k} + \Gamma^k\Phi(e_x - \pi)$$

where $\Gamma^k = E^\dagger\beta$. After choosing a variance criterion our solution for Γ^k can be expressed directly as a function of the covariance of the embedding V and its derivative with respect to the probability vector, $\frac{dV}{d\pi_k}$. Consequently, $S(x)$ can be expressed directly as a function of the probability vector, $\pi(\eta) \in \Delta^{K-1}$ and the embedding matrix, Φ.

In the following analysis we seek to solve for the matrix Γ^k under two conditions. We first consider the most natural criterion: the total variance case where $E = I$. We then extend the analysis to the case where $E = V^\dagger$, yielding a particularly simple solution that recovers the Zero-Gumbel-Rao (ZGR) estimator.

Throughout this analysis we denote $\frac{\partial V}{\partial p_k}$ as C^k for notational convenience.

B.3 Case $E = I$.

If $E = I$, the equation becomes

$$V\beta + \beta^\top V = C^k. \tag{71}$$

We also note that E has full rank and therefore our constraints and solution lie entirely within the range of E. We can solve for the symmetric solution β using the eigen-decomposition $V = U\Lambda U^\top$:

$$U\Lambda U^\top\beta + \beta U\Lambda U^\top = C^k \tag{72}$$

$$\Lambda U^\top\beta U + U^\top\beta U\Lambda = U^\top C^k U \tag{73}$$

$$\Lambda\tilde{\beta} + \tilde{\beta}\Lambda = \tilde{C}^k, \tag{74}$$

with $\tilde{\beta} = U^\top\beta U$ and $\tilde{C}^k = U^\top C^k U$. Since V is positive semi-definite, the singularity $\lambda_i + \lambda_j = 0$ occurs only when both eigenvalues are zero. In this subspace, the derivative $\tilde{C}^k$ vanishes automatically, guaranteeing a solution without further assumptions. Then

$$\beta = U\tilde{\beta}U^\top, \quad \tilde{\beta}_{ij} = \tfrac{\tilde{C}^k_{ij}}{\lambda_i+\lambda_j}. \tag{75}$$

It follows that

$$\Gamma^k = U\tilde{\beta}U^\top \tag{76}$$

$$\Gamma^k = \textstyle\sum_{i,j}\left(\frac{\tilde{C}^k_{ij}}{\lambda_i+\lambda_j}\right) u_i u_j^\top. \tag{77}$$

While naively calculating Γ^k for each k independently yields a total complexity of $O(Kd^3)$, we demonstrate that the design matrix S may be evaluated in $O(Kd^2)$

$$S(x)_{:k} = \Phi_{:k} + \textstyle\sum_{i,j}\left(\frac{\tilde{C}^k_{ij}}{\lambda_i+\lambda_j}\right) u_i u_j^\top \Phi(e_x - \pi) \tag{78}$$

$$S(x)_{:k} = \Phi_{:k} + \textstyle\sum_{i,j}\left(\frac{u_i^\top C^k u_j}{\lambda_i+\lambda_j}\right) u_i u_j^\top \Phi(e_x - \pi) \tag{79}$$

$$S(x)_{:k} = \Phi_{:k} + \textstyle\sum_{i,j}\left(\frac{u_i^\top(\Phi_{:k}\Phi_{:k}^\top - \Phi_{:k}M^\top - M\Phi_{:k}^\top)u_j}{\lambda_i+\lambda_j}\right) u_i u_j^\top \Phi(e_x - \pi) \tag{80}$$

$$S(x)_{:k} = \Phi_{:k} + \textstyle\sum_{i,j}\frac{u_j^\top\Phi(e_x-\pi)}{\lambda_i+\lambda_j} u_i\left(u_i^\top\left(\Phi_{:k}\Phi_{:k}^\top - \Phi_{:k}M^\top - M\Phi_{:k}^\top\right)u_j\right) \tag{81}$$

$$S(x) = \Phi + \textstyle\sum_{i,j}\left(\frac{u_j^\top\Phi(e_x-\pi)}{\lambda_i+\lambda_j}\right) u_i\Big[(u_i^\top\Phi)\mathrm{diag}(u_j^\top\Phi) - (u_j^\top M)(u_i^\top\Phi) - (u_i^\top M)(u_j^\top\Phi)\Big] \tag{82}$$

$$\tag{83}$$

Evaluating this expression for the design matrix $S(x)$ has time complexity $O(d^2K)$ and we note that the eigen-decomposition of V has complexity $O(d^3)$ and $d \leq K$.

B.4 Case $E = V^\dagger$, $\Phi = I$.

In this case, E is rank-deficient and therefore our solution has a component in the null space of E. Our equation for β can be written as

$$V\beta V + V\beta^\top V = \Pi_V(C) \tag{84}$$

where Π_V is the orthogonal projector onto the range of V and our estimator can be written as

$$S(x)_{:k} = \Phi_{:k} + \Gamma^k\Phi(e_x - \pi) + N(x) \tag{85}$$

where $N(x)$ is the component in the null space of E. The null space component must satisfy the following null space constraints:

$$\mathbb{E}[N(x)] = 0 \tag{86}$$
$$\textstyle\sum_x p(x) \left[\phi(x)_i N(x)_{jk} + \phi(x)_j N(x)_{ik}\right] = \Pi_{\mathcal{N}(V)}(C) \tag{87}$$

Our symmetric solution β is given by

$$\beta = \tfrac{1}{2} V^\dagger \Pi_V(C) V^\dagger. \tag{88}$$

It follows that

$$\Gamma^k = \tfrac{1}{2}(V V^\dagger \Pi_V(C) V^\dagger) \tag{89}$$

Recognizing that $VV^\dagger$ acts as a projection onto the range of V, we apply the idempotence of projections to simplify the result

$$\Gamma^k = \tfrac{1}{2}(\Pi_V(C) V^\dagger) \tag{90}$$

We observe that in the one-hot case $\text{Null}(V) \subseteq \text{Null}(C)$ and therefore our expression simplifies as

$$\Gamma^k = \tfrac{1}{2} C V^\dagger \tag{91}$$

Furthermore $\Pi_{\mathcal{N}(V)}(C) = 0$ and therefore $N(x) = 0$ is a valid solution. Substituting our two solutions into the expression for the design matrix yields

$$S(x)_{:k} = e_k + \tfrac{1}{2} C V^\dagger (e_x - \pi) \tag{92}$$

Expanding the covariance matrix of the embedding in the case of the one-hot embedding we obtain

$$S(x)_{:k} = e_k + \tfrac{1}{2}(e_k e_k^\top - e_k \pi^\top - \pi e_k^\top) V^\dagger (e_x - \pi) \tag{93}$$

Expanding the pseudo-inverse $V^\dagger$ in the case of the one-hot embedding gives us

$$\begin{aligned} V^\dagger &= (\text{diag}(\pi))^{-1} + \tfrac{\sum_{i=1}^n \frac{1}{\pi_i}}{n^2} \mathbf{1}\mathbf{1}^T - \tfrac{1}{n}\left((\text{diag}(\pi))^{-1}\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T(\text{diag}(\pi))^{-1}\right) \\ &= (\text{diag}(\pi))^{-1} + F\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T G. \end{aligned} \tag{94}$$

The product terms involving F and G will effectively vanish because:

$$F 1 1^\top (e_x - \pi) = F 1 (1 - 1) = 0 \tag{95}$$

and

$$\begin{aligned} &(e_k e_k^\top - e_k \pi^\top - \pi e_k^\top)\mathbf{1}\mathbf{1}^\top G(e_x - \pi) \\ &= \left[\mathbf{1}^\top G(e_x - \pi)\right](e_k e_k^\top - e_k \pi^\top - \pi e_k^\top)\mathbf{1} \\ &= -\left[\mathbf{1}^\top G(e_x - \pi)\right]\pi \end{aligned} \tag{96}$$

The second term in identical across all columns and is therefore equal to zero after multiplying by the Jacobian. Substituting $V^\dagger$ back into the expression for the design matrix we obtain

$$\begin{aligned} S(x)_{:k} &= I_{:k} + \tfrac{1}{2}(e_k e_k^\top - e_k \pi^\top - \pi e_k^\top)(\mathrm{diag}(\pi))^{-1}(e_x - \pi) \\ &= e_k + \tfrac{1}{2}(e_k e_k^\top - e_k \pi^\top - \pi e_k^\top)\left(\tfrac{e_x}{p(x)} - \mathbf{1}\right). \end{aligned} \tag{97}$$

After expanding the brackets we notice that the second term effectively vanishes as before. The final expression simplifies as follows

$$\begin{aligned} S(x)_{:k} &= e_k + \tfrac{1}{2}(e_k e_k^\top - e_k \pi^\top - \pi e_k^\top)\tfrac{e_x}{p(x)} \\ &= e_k + \tfrac{1}{2}\left(\tfrac{1}{p(x)} e_k e_k^\top e_x - e_k - \tfrac{\pi e_k^\top e_x}{p(x)}\right) \\ &= \tfrac{1}{2}\Big(e_k + \tfrac{1}{p(x)}\,(e_x - \pi)\,\delta_{k=x}\Big). \end{aligned} \tag{98}$$

We obtain the full design matrix

$$S(x) = \frac{1}{2}\Big(I + \frac{1}{p(x)}\,(e_x - \pi)\,e_x^\top\Big)$$

The full gradient estimator in η can be expressed as

$$\begin{aligned} \hat{J}_\eta &= \tfrac{1}{2}\tfrac{d\mathcal{L}}{d\phi}\Phi\Big(\tfrac{d\pi}{d\eta} + \tfrac{1}{p(x)}(e_x - \pi)\tfrac{d\pi_x}{d\eta}\Big) \\ &= \tfrac{1}{2}\tfrac{d\mathcal{L}}{d\phi}\Phi\Big(\tfrac{d\pi}{d\eta} + (e_x - \pi)\tfrac{d\log p(x;\eta)}{d\eta}\Big) \\ &= \tfrac{1}{2}\tfrac{d\mathcal{L}}{d\phi}\Big(\tfrac{d\bar{\phi}}{d\eta} + (\phi(x) - \bar{\phi})\tfrac{d\log p(x;\eta)}{d\eta}\Big), \end{aligned} \tag{99}$$

which is precisely the ZGR estimator. We have therefore shown that ZGR is equivalent to MVE with the $E = V^\dagger$ variance criterion for the one-hot embedding.

C MINIMUM TOTAL VARIANCE ESTIMATOR WITH A QUADRATIC BIAS PENALTY FUNCTION

We restrict our attention to the case where E has full rank and modify the Lagrangian by replacing the hard linear and quadratic constraints with soft penalty terms. We proceed by taking the limit $\rho \to \infty$, converting the linear penalty into a hard constraint. The resulting family of solutions corresponds to minimum-variance estimators with a fixed allowance for quadratic bias, parameterized by the quadratic penalty coefficient.

In the limit as the quadratic penalty coefficient tends to infinity, the estimator becomes unbiased for quadratic functions, while setting the penalty coefficient to zero recovers the straight-through estimator. We note that although softening the linear constraint before taking the limit may appear redundant, this relaxation significantly simplifies the calculations.

$$\begin{aligned} L_{\rho,\tau}(S) =& \mathbb{E}_{a,x}\left[\left(a^\top S(x) e_k\right)^2\right] \\ &+ \tfrac{\tau}{2}\textstyle\sum_{ij}\left(\Phi_{ik}\Phi_{jk} - \sum_x p(x)\left[\Phi_{ix}S(x)_{jk} + \Phi_{jx}S(x)_{ik}\right]\right)^2 \\ &+ \tfrac{\rho}{2}\textstyle\sum_j\left(\Phi_{jk} - \sum_x p(x) S(x)_{jk}\right)^2 \end{aligned}$$

Differentiating with respect to $S(x)_{jk}$ yields:

$$\begin{aligned}\tfrac{\partial L}{\partial S(x)_{jk}} &= 2p(x)\,\mathbb{E}_a\big[a_j\,\big(a^\top S(x)e_k\big)\big]\\ &\quad - 2\tau p(x)\textstyle\sum_i \Phi_{ix}\,(\Phi_{ik}\Phi_{jk} - \sum_{x'} p(x')\,[\Phi_{ix'}S(x')_{jk} + \Phi_{jx'}S(x')_{ik}])\\ &\quad - \rho p(x)\,(\Phi_{jk} - \textstyle\sum_{x'} p(x')S(x')_{jk})\,.\end{aligned} \tag{100}$$

Collecting the derivatives over the index j gives the vector-valued condition:

$$\begin{aligned}\tfrac{\partial L}{\partial S(x)_{:k}} &= 2p(x)\,ES(x)_{:k}\\ &\quad - 2\tau p(x)\textstyle\sum_i \Phi_{ix}\,(\Phi_{ik}\Phi_{:k} - \sum_{x'} p(x')\,[\Phi_{ix'}S(x')_{:k} + \Phi_{:x'}S(x')_{ik}])\\ &\quad - \rho p(x)\,(\Phi_{:k} - \textstyle\sum_{x'} p(x')S(x')_{:k})\,.\end{aligned} \tag{101}$$

Setting the gradient equal to zero and rearranging terms yields:

$$\begin{aligned}&ES(x)_{:k} + \tau\textstyle\sum_i \Phi_{ix}\sum_{x'} p(x')\,[\Phi_{ix'}S(x')_{:k} + \Phi_{:x'}S(x')_{ik}] + \tfrac{\rho}{2}\textstyle\sum_{x'} p(x')S(x')_{:k}\\ &\quad = \big(\tau\textstyle\sum_i \Phi_{ix}\Phi_{ik} + \tfrac{\rho}{2}\big)\,\Phi_{:k}.\end{aligned} \tag{102}$$

Viewing $S(x)_{jk}$ as a tensor S_{jxk}, we fix the index k and work with the matrix S_{jx}:

$$\begin{aligned}&ES_{:x} + \tau\textstyle\sum_i \Phi_{ix}\sum_{x'} p(x')\,[\Phi_{ix'}S_{:x'} + \Phi_{:x'}S_{ix'}] + \tfrac{\rho}{2}\textstyle\sum_{x'} p(x')S_{:x'}\\ &\quad = \big(\tau\textstyle\sum_i \Phi_{ix}\Phi_{ik} + \tfrac{\rho}{2}\big)\,\Phi_{:k}.\end{aligned} \tag{103}$$

Rearranging the second term and writing p in index form yields:

$$\begin{aligned}&ES_{:x} + \tau\textstyle\sum_{i,x'} p_{x'}\,[\Phi_{ix}\Phi_{ix'}S_{:x'} + \Phi_{:x'}\Phi_{ix}S_{ix'}] + \tfrac{\rho}{2}\textstyle\sum_{x'} p_{x'}S_{:x'}\\ &\quad = \big(\tau\textstyle\sum_i \Phi_{ix}\Phi_{ik} + \tfrac{\rho}{2}\big)\,\Phi_{:k}.\end{aligned} \tag{104}$$

Summing over the index i yields:

$$\begin{aligned}&ES_{:x} + \tau\textstyle\sum_{x'} p_{x'}\,\big[(\Phi^\top\Phi)_{xx'}S_{:x'} + \Phi_{:x'}(\Phi^\top S)_{xx'}\big] + \tfrac{\rho}{2}\textstyle\sum_{x'} p_{x'}S_{:x'}\\ &\quad = \big(\tau(\Phi^\top\Phi)_{xk} + \tfrac{\rho}{2}\big)\,\Phi_{:k}.\end{aligned} \tag{105}$$

Summing over the index i gives:

$$\begin{aligned}&ES_{:x} + \tau\textstyle\sum_{x'} p_{x'}\,\big[(\Phi^\top\Phi)_{xx'}S_{:x'} + \Phi_{:x'}(\Phi^\top S)_{xx'}\big] + \tfrac{\rho}{2}\textstyle\sum_{x'} p_{x'}S_{:x'}\\ &\quad = \big(\tau(\Phi^\top\Phi)_{xk} + \tfrac{\rho}{2}\big)\,\Phi_{:k}.\end{aligned} \tag{106}$$

Recognizing the sums over x' as matrix products yields:

$$\begin{aligned}&ES_{:x} + \tau\big[S\,\mathrm{diag}(p)\Phi^\top\Phi\big]_{:x} + \tau\big[\Phi\,\mathrm{diag}(p)S^\top\Phi\big]_{:x} + \tfrac{\rho}{2}Sp\\ &\quad = \big(\tau(\Phi^\top\Phi)_{xk} + \tfrac{\rho}{2}\big)\,\Phi_{:k}.\end{aligned} \tag{107}$$

Expanding the right-hand side and rewriting the system as a matrix equation in $\mathcal{S} = \{S_{jx}\}$ gives:

$$\begin{aligned}&E\mathcal{S} + \tau\mathcal{S}\,\mathrm{diag}(p)\Phi^\top\Phi + \tau\Phi\,\mathrm{diag}(p)\mathcal{S}^\top\Phi + \tfrac{\rho}{2}\mathcal{S}p\mathbf{1}^\top\\ &\quad = \tau\Phi_{:k}(\Phi^\top\Phi)_{k:} + \tfrac{\rho}{2}\Phi_{:k}\mathbf{1}^\top.\end{aligned} \tag{108}$$

We now decompose the left-hand side into components orthogonal to, and aligned with, the subspace of matrices with constant rows. Writing $v\mathbf{1}^\top$ for the projection onto this subspace, we obtain:

$$E\mathcal{S} + \tau\mathcal{S}\,\mathrm{diag}(p)\Phi^\top\Phi + \tau\Phi\,\mathrm{diag}(p)\mathcal{S}^\top\Phi = \tau\Phi_{:k}\Phi_{:k}^\top\Phi + v\mathbf{1}^\top, \tag{109}$$

$$v\mathbf{1}^\top + \tfrac{\rho}{2}(\mathcal{S}p)\mathbf{1}^\top = \Phi_{:k}\left(\tau\Phi_{:k}^\top\Phi + \tfrac{\rho}{2}\mathbf{1}^\top\right). \tag{110}$$

Finally, taking the limit $\rho \to \infty$ enforces the linear constraint, yielding:

$$\boxed{\begin{aligned} E\mathcal{S} + \tau\mathcal{S}\,\mathrm{diag}(p)\Phi^\top\Phi + \tau\Phi\,\mathrm{diag}(p)\mathcal{S}^\top\Phi &= \tau\Phi_{:k}\Phi_{:k}^\top\Phi + v\mathbf{1}^\top \\ \mathcal{S}p &= \Phi_{:k} \end{aligned}} \tag{111}$$

This is a system of $d(K+1)$ variables in $d(K+1)$ variables $\mathcal{S}, v$. Since only the RHS depends on K, the complexity of solving it for all k via inverting the matrix of size $(dK) \times (dK)$ is $O((dK)^3)$.

C.1 Special Cases

Working from (111), we restrict our attention to two special cases. First, we present an analytic solution for the 1D embedding. Second, we characterize the one-hot embedding under the total variance criterion ($E = I$), where a closed-form expression is obtainable. The more general solution, involving an arbitrary embedding, is deferred to later work.

C.1.1 1D Embedding

Proposition C.1. *The solution to the Lagrangian subproblem for the 1D embedding ϕ is the Minimum Variance Estimator with design matrix*

$$S(x) = \frac{(1 - 2\tau m_1 x + 2\tau m_2)\Phi + \tau(x - m_1)(\Phi \odot \Phi)}{1 + 2\tau V}. \tag{112}$$

Proof. In the 1D embedding case, we may exploit the fact that the embedding matrix is a vector and its first dimension is equal to one. This allows the matrix-valued optimality conditions to be reduced to scalar equations, which are significantly easier to solve analytically.

We begin by right-multiplying the first equation of the boxed system by $\mathrm{diag}(p)\Phi^\top$, obtaining

$$\begin{aligned} &S\,\mathrm{diag}(p)\Phi^\top + \tau S\,\mathrm{diag}(p)\Phi^\top\Phi\,\mathrm{diag}(p)\Phi^\top + \tau\Phi\,\mathrm{diag}(p)S^\top\,\mathrm{diag}(p)\Phi^\top\Phi \\ &\quad = \tau\Phi_{:k}\Phi_{:k}^\top\Phi\,\mathrm{diag}(p)\Phi^\top + v\mathbf{1}^\top\,\mathrm{diag}(p)\Phi^\top. \end{aligned} \tag{113}$$

Noticing the following identities:

$$m_1 = \mathbf{1}^\top\,\mathrm{diag}(p)\Phi^\top, \qquad m_2 = \Phi\,\mathrm{diag}(p)\Phi^\top, \tag{114}$$

and introducing the variable

$$y = S\,\mathrm{diag}(p)\Phi^\top, \tag{115}$$

the system reduces to the scalar equation

$$y + 2\tau y m_2 = \tau\Phi_k^2 m_2 + m_1 v. \tag{116}$$

Solving for y yields

$$y = \frac{\tau\Phi_k^2 m_2 + m_1 v}{1 + 2\tau m_2}. \tag{117}$$

Substituting y into the original matrix equation isolates S in terms of v:

$$S + 2\tau y\Phi = \tau\Phi_k^2\Phi + v\mathbf{1}^\top. \tag{118}$$

$$S = (\tau\Phi_k^2 - 2\tau y)\Phi + v\mathbf{1}^\top, \tag{119}$$

and substituting in our value for y yields

$$S = \tau\left(\frac{\Phi_k^2 - 2m_1 v}{1 + 2\tau m_2}\right)\Phi + v\mathbf{1}^\top. \tag{120}$$

To solve for v, we multiply by p and substitute the linear constraint from our original pair of equations, giving

$$\Phi_k = \tau \left(\frac{\Phi_k^2 - 2m_1 v}{1 + 2\tau m_2} \right) m_1 + v. \tag{121}$$

Solving for v yields

$$v = \frac{\Phi_k(1 + 2\tau m_2) - \tau m_1 \Phi_k^2}{1 + 2\tau V}. \tag{122}$$

Substituting v into our equation for S and simplifying gives

$$S = \tau \left(\frac{\Phi_k^2 - 2m_1 \frac{\Phi_k(1+2\tau m_2) - \tau m_1 \Phi_k^2}{1+2\tau V}}{1+2\tau m_2} \right) \Phi + \frac{\Phi_k(1+2\tau m_2) - \tau m_1 \Phi_k^2}{1+2\tau V} \mathbf{1}^\top, \tag{123}$$

$$S = \tau \left(\frac{\Phi_k^2}{1+2\tau m_2} - 2m_1 \frac{\Phi_k(1+2\tau m_2) - \tau m_1 \Phi_k^2}{(1+2\tau V)(1+2\tau m_2)} \right) \Phi + \frac{\Phi_k(1+2\tau m_2) - \tau m_1 \Phi_k^2}{1+2\tau V} \mathbf{1}^\top, \tag{124}$$

$$S = \tau \left(\frac{(1+2\tau V)\Phi_k^2 + 2\tau m_1^2 \Phi_k^2 - 2m_1 \Phi_k (1+2\tau m_2)}{(1+2\tau V)(1+2\tau m_2)} \right) \Phi + \frac{\Phi_k(1+2\tau m_2) - \tau m_1 \Phi_k^2}{1+2\tau V} \mathbf{1}^\top, \tag{125}$$

$$S = \tau \left(\frac{\Phi_k^2 - 2m_1 \Phi_k}{1+2\tau V} \right) \Phi + \frac{\Phi_k(1+2\tau m_2) - \tau m_1 \Phi_k^2}{1+2\tau V} \mathbf{1}^\top. \tag{126}$$

Re-indexing the tensor back into functional form and simplifying yields

$$S_k(x) = \tau \left(\frac{\Phi_k^2 - 2m_1 \Phi_k}{1 + 2\tau V} \right) x + \frac{\Phi_k(1 + 2\tau m_2) - \tau m_1 \Phi_k^2}{1 + 2\tau V}, \tag{127}$$

$$S_k(x) = \frac{(1 - 2\tau m_1 x + 2\tau m_2)\Phi_k + \tau(x - m_1)\Phi_k^2}{1 + 2\tau V}. \tag{128}$$

Finally, for general k, we obtain

$$S(x) = \frac{(1 - 2\tau m_1 x + 2\tau m_2)\Phi + \tau(x - m_1)(\Phi \odot \Phi)}{1 + 2\tau V}. \tag{129}$$

□

C.1.2 One-Hot Embedding

Proposition C.2. *The solution to the Lagrangian subproblem for the one-hot embedding ϕ and total variance criterion is the Minimum Variance Estimator (***MVE***) with design matrix*

$$S(x) = M_1 M_2 A^{-1} M_3 + M_4, \tag{130}$$

where the component matrices are defined as:

$$M_1 = \operatorname{diag}\left(\tfrac{\tau}{1+\tau(p+p(x))} \right), \qquad M_3 = \operatorname{diag}\left(\tfrac{1+\tau p}{1+2\tau p} \right),$$
$$M_2 = \operatorname{diag}\left(\tfrac{1}{\tau} + p \right) - p\, e_x^\top, \qquad M_4 = \tfrac{\tau}{1+2\tau p(x)}\, e_x e_x^\top,$$

and the matrix A is given by

$$A_{ij} = \delta_{ij}\, d_i - \tfrac{\tau p_i p_j}{1+\tau p_i + \tau p_j}, \qquad d_i = \textstyle\sum_j \tfrac{(1+\tau p_i)\, p_j}{1+\tau(p_i+p_j)}. \tag{131}$$

Proof. In the case of the One-Hot embedding, the embedding matrix is equal to the identity matrix. This simplifies our pair of equations significantly. Unfortunately, we are still unable to solve for a general variance matrix, but we may seek an analytical solution in the total variance case. Our pair of equations simplify to:

$$S(\tau \operatorname{diag}(p) + I) + \tau \operatorname{diag}(p) S^\top = \tau e_k e_k^\top + \mathbf{v}\mathbf{1}^\top, \tag{132}$$
$$S\mathbf{p} = e_k. \tag{133}$$

The first equation is now a Sylvester-transpose equation with diagonal coefficients, and therefore has the following analytical solution:

$$S_{ijk} = \frac{(1+\tau p_i)v_{ik} - \tau p_i v_{jk} + \tau \delta_{ij}\delta_{ik}}{\tau(p_i+p_j)+1}. \tag{134}$$

Substituting this solution into the linear equation (133) gives

$$\sum_j \frac{(\tau p_i+1)v_{ik} - (\tau p_i)v_{jk} + \tau\delta_{ij}\delta_{ik}}{\tau p_i + \tau p_j + 1}\, p_j = \delta_{ik}. \tag{135}$$

We break the sum into three terms depending on the indexing of v:

$$\sum_j \frac{(\tau p_i+1)v_{ik}p_j}{\tau p_i+\tau p_j+1} - \sum_j \frac{\tau p_i v_{jk} p_j}{\tau p_i+\tau p_j+1} + \sum_j \frac{\tau\delta_{ij}\delta_{ik}p_j}{\tau p_i+\tau p_j+1} = \delta_{ik}. \tag{136}$$

Rearranging into a linear system in terms of v gives

$$(\tau p_i+1)v_{ik}\sum_j \frac{p_j}{\tau p_i+\tau p_j+1} - \tau p_i \sum_j \frac{v_{jk}p_j}{\tau p_i+\tau p_j+1} = \frac{\delta_{ik}(\tau p_i+1)}{2\tau p_i+1}. \tag{137}$$

This can be written compactly as $AV = B$, where

$$B = (\tau\,\mathrm{diag}(p) + I)\,(I + 2\tau\,\mathrm{diag}(p))^{-1}, \tag{138}$$

$$A_{ij} = \delta_{ij}d_i - \tfrac{\alpha_i\beta_j}{x_i+y_j}, \tag{139}$$

with

$$d_i = \sum_j \frac{(1+\tau p_i)\,p_j}{1+\tau(p_i+p_j)}, \quad \alpha_i = \tau p_i, \quad \beta_j = p_j, \quad x_i = 1+\tau p_i, \quad y_j = \tau p_j. \tag{140}$$

Solving for v and writing in elementwise form gives

$$v_{ik} = (A^{-1})_{ik}\frac{\tau p_k+1}{2\tau p_k+1}. \tag{141}$$

We now substitute our solution for v into the equation for S:

$$S_{ijk} = \frac{\tau p_k+1}{(2\tau p_k+1)\,[\tau(p_i+p_j)+1]}\Big[(1+\tau p_i)(A^{-1})_{ik} - \tau p_i (A^{-1})_{jk}\Big] + \frac{\tau\delta_{ij}\delta_{ik}}{\tau(p_i+p_j)+1}. \tag{142}$$

Reindexing the tensor back into functional form gives

$$S_{ik}(x) = \frac{\tau p_k+1}{(2\tau p_k+1)\,[\tau(p_i+p_x)+1]}\Big[(1+\tau p_i)(A^{-1})_{ik} - \tau p_i (A^{-1})_{xk}\Big] + \frac{\tau\delta_{ix}\delta_{ik}}{\tau(p_i+p_x)+1}. \tag{143}$$

Finally, this can be written compactly as

$$\boxed{S(x) = \mathrm{diag}\left(\tfrac{\tau}{1+\tau(p+p(x))}\right)\Big(\mathrm{diag}(\tfrac{1}{\tau}+p) - p\,e_x^\top\Big)A^{-1}\,\mathrm{diag}\left(\tfrac{1+\tau p}{1+2\tau p}\right) + \tfrac{\tau}{1+2\tau p(x)}\,e_x e_x^\top} \tag{144}$$

This obtains the quadratically unbiased minimum variance estimator for one-hot categorical variables.

Unfortunately, A has no analytical inverse. However, it is a highly structured matrix—specifically, the sum of a diagonal and a Cauchy-like matrix. While this structure suggests an efficient inverse exists, our algorithm simply exploits its symmetry via a standard Cholesky inverse.

□

C.2 Temperature Limits

By examining the limiting behavior of our parametric family with respect to temperature, we can recover specific estimators of interest. Specifically, the limit as temperature approaches infinity yields the quadratically unbiased estimator for the chosen embedding and variance criterion. Conversely, setting the temperature to zero recovers the straight-through estimator, regardless of the embedding or variance criterion. We begin our analysis with the 1D embedding case:

$$\begin{aligned} S_0(x) &= \tfrac{(1)\Phi+(0)(\Phi\odot\Phi)}{1} \\ &= \Phi. \end{aligned} \tag{145}$$

Proposition C.3. *For the 1D embedding, the minimum-variance (w.r.t.* (12) *with any non-degenerate distribution) quadratically unbiased GST estimator is given by*

$$\mathbf{S}(x) = \alpha(x)\Phi + \beta(x)\Phi \odot \Phi, \tag{146}$$

where:

$$\alpha(x) = \tfrac{m_2-\phi(x)m_1}{v}, \quad \beta(x) = \tfrac{\phi(x)-m_1}{2v} \tag{147}$$

and the underlying moments and variance are, respectively, $m_1 = \mathbb{E}[\phi(x)]$, $m_2 = \mathbb{E}[\phi(x)^2]$, $v = \mathbb{V}[\phi(x)] = m_2 - m_1^2$.

Proof. For the infinite temperature limit, we have:

$$\begin{aligned} \lim_{\tau\to\infty} S_\tau(x) &= \lim_{\tau\to\infty} \tfrac{(1-2\tau m_1 x+2\tau m_2)\Phi+\tau(x-m_1)(\Phi\odot\Phi)}{1+2\tau V} \\ &= \tfrac{2(m_2-m_1x)\Phi+(x-m_1)(\Phi\odot\Phi)}{2V} \\ &= \left(\tfrac{m_2-m_1x}{V}\right)\Phi + \left(\tfrac{x-m_1}{2V}\right)(\Phi\odot\Phi). \end{aligned} \tag{148}$$

□

We now consider the case of the one-hot embedding at zero temperature:

$$\begin{aligned} [A_{ij}]_{\tau=0} &= \left[\delta_{ij}\textstyle\sum_k \tfrac{p_k+\tau p_i p_k}{1+\tau(p_i+p_k)} - \tfrac{\tau p_i p_j}{1+\tau(p_i+p_j)}\right]_{\tau=0} \\ &= \delta_{ij}\textstyle\sum_k p_k \\ &= \delta_{ij} \\ \implies A &= I. \end{aligned} \tag{149}$$

Substituting this back into the estimator:

$$\begin{aligned} [S(x)]_{\tau=0} &= \left[\operatorname{diag}\left(\tfrac{1}{1+\tau(p+p(x))}\right)\Big(\operatorname{diag}(1+\tau p) - \tau p\, e_x^\top\Big)A^{-1}\operatorname{diag}\left(\tfrac{1+\tau p}{1+2\tau p}\right) + \tfrac{\tau}{1+2\tau p(x)}\, e_x e_x^\top\right]_{\tau=0} \\ &= I\cdot(I-0)\cdot I\cdot I + 0 \\ &= I \\ &= \Phi. \end{aligned} \tag{150}$$

Proposition C.4. *The quadratically unbiased GST estimator with the total variance criterion* (13) *can be expressed as follows. Let* $D(x) = \operatorname{diag}\left(\tfrac{1}{p+p(x)}\right)$, *and define the matrix* $A(x) = \operatorname{diag}(p) - p\, e_x^\top$. *Furthermore, let* L *be the matrix with elements* L_{ij} *given by:*

$$L_{ij} = \begin{cases} \sum_{k\neq i} \tfrac{p_i p_k}{p_i+p_k} & \text{if } i = j, \\ -\tfrac{p_i p_j}{p_i+p_j} & \text{if } i \neq j. \end{cases} \tag{151}$$

Then, the estimator is given by:

$$S(x) = \frac{1}{2}\left[D(x)A(x)L^\dagger + \frac{1}{p(x)}\, e_x e_x^\top\right]. \tag{152}$$

Proof. Next, we examine the limit as the temperature approaches infinity ($\tau \to \infty$). We begin by computing the limit of the A matrix:

$$\lim_{\tau\to\infty} A_{ij} = \begin{cases} \sum_{k\neq i} \frac{p_i p_k}{p_i+p_k} & \text{if } i = j, \\ -\frac{p_i p_j}{p_i+p_j} & \text{if } i \neq j. \end{cases} \tag{153}$$

Computing the full limit and denoting the limiting matrix as L, we obtain:

$$S(x) = \frac{1}{2}\left[\operatorname{diag}\Big(\frac{1}{p+p(x)}\Big)\Big(\operatorname{diag}(p) - p\, e_x^\top\Big)L^\dagger + \frac{1}{p(x)}\, e_x e_x^\top\right]. \tag{154}$$

We note here that the matrix L is of the form of a weighted graph Laplacian. □

D UNBIASED SQUARED BIAS AND VARIANCE OF A STOCHASTIC ESTIMATOR

Let X be a random variable with discrete distribution $p(x)$, $x \in \{1, \dots, K\}$. Let $f(X, Y)$ be a function of X and another random variable Y not independent from X. For each X we have n conditional samples of Y and we collect $f_{k,i} = f(X = k, Y_i)$.

Problem 1: Unbiased estimate of $\mathbb{E}[f(X, Y)]^2$.

Let us denote the mean $\mu = \mathbb{E}[f(X, Y)]$, the second moment $M = \mathbb{E}[f(X, Y)^2]$ and the conditional moments:

$$\mu_k = \mathbb{E}[f(X = k, Y)], \quad M_k = \mathbb{E}[f(X = k, Y)^2]. \tag{155}$$

We can compute unnbiased estimates of these quantities:

$$\hat{\mu}_k = \tfrac{1}{n}\textstyle\sum_{i=1}^n f_{k,i}, \quad \hat{M}_k = \tfrac{1}{n}\textstyle\sum_{i=1}^n f_{k,i}^2. \tag{156}$$

We can also compute the overall mean and second moment:

$$\hat{\mu} = \textstyle\sum_k p(k)\hat{\mu}_k, \quad \hat{M} = \textstyle\sum_k p(k)\hat{M}_k. \tag{157}$$

We have $\mathbb{E}[\hat{\mu}] = \mu$ and $\mathbb{E}[\hat{M}] = M$. If the conditional sample batches $\{Y_{k,i}\}_{i=1}^n$ and $\{Y_{l,i}\}_{i=1}^n$ are independent for all $k \neq l$, then the empirical means $\hat{\mu}_k = \frac{1}{n}\sum_{i=1}^n f(k, Y_{k,i})$ are mutually independent; hence for $k \neq l$, $\mathbb{E}[\hat{\mu}_k\hat{\mu}_l] = \mu_k\mu_l$.

However, $\mathbb{E}[\hat{\mu}^2] \neq \mu^2$. Let us express it and correct the bias:

$$\mathbb{E}[\hat{\mu}^2] = \mathbb{E}\left[\left(\textstyle\sum_k p(k)\hat{\mu}_k\right)^2\right] = \textstyle\sum_k p(k)^2\mathbb{E}[\hat{\mu}_k^2] + \sum_{k\neq l} p(k)p(l)\mathbb{E}[\hat{\mu}_k\hat{\mu}_l] \tag{158}$$

$$= \textstyle\sum_k p(k)^2\left(\mathbb{V}(\hat{\mu}_k) + \mathbb{E}[\hat{\mu}_k]^2\right) + \sum_{k\neq l} p(k)p(l)\mathbb{E}[\hat{\mu}_k]\mathbb{E}[\hat{\mu}_l] \tag{159}$$

$$= \textstyle\sum_k p(k)^2\left(\frac{\mathbb{V}(f(X=k,Y))}{n} + \mu_k^2\right) + \sum_{k\neq l} p(k)p(l)\mu_k\mu_l \tag{160}$$

$$= \textstyle\sum_k p(k)^2\frac{M_k-\mu_k^2}{n} + \mu^2. \tag{161}$$

An unbiased estimate of μ_k^2 can be computed as follows:

$$\hat{\mu}_k^2 - \tfrac{\hat{M}_k - \hat{\mu}_k^2}{n-1}. \tag{162}$$

Thus, an unbiased estimate of μ^2 is:

$$\tilde{B} = \hat{\mu}^2 - \textstyle\sum_k p(k)^2\Big(\hat{M}_k - \Big(\hat{\mu}_k^2 - \frac{\hat{M}_k-\hat{\mu}_k^2}{n-1}\Big)\Big)\frac{1}{n} = \hat{\mu}^2 - \sum_k p(k)^2\frac{\hat{M}_k-\hat{\mu}_k^2}{n-1}. \tag{163}$$

Problem 2: Unbiased estimate of $\mathbb{V}[f(X, Y)]$.

Using $\mathbb{V}[f(X, Y)] = M - \mu^2$, an unbiased estimate of the variance is

$$\tilde{V} = \hat{M} - \tilde{B}. \tag{164}$$

E DETAILS OF EXPERIMENTS

We evaluated the following gradient estimators:

- **MVE** - the proposed Minimum Variance Estimator using Total Variance Proxy Criterion (1D or Categorical)
- **MVE**(τ) - the proposed parametric Minimum Variance Estimator using Total Variance Proxy Criterion (1D or Categorical)
- **GRMC** - Gumbel-Rao Monte Carlo estimator (Paulus et al., 2021) with $M = 10$, 20 and 100 samples.
- **STGS** - Straight-Through Gumbel-Softmax estimator (Jang et al., 2017) implemented as GRMC with $M = 1$ sample.
- **ZGR** - Zero temperature Gumbel-Rao estimator (5) (Shekhovtsov, 2023).
- **ST** - the Straight-Through estimator (2).

We implemented all methods in Pytorch.

E.1 Bias-Variance Tradeoff

Fig. E.1 illustrates the bias-variance tradeoff for different gradient estimators across categorical dimensions K and random seeds. For each plot we draw a random probability distribution $p(x)$ and a random quadratic loss function $\ell(\phi) = \|L\phi - c\|_2^2$ on R^K once. We then enumerate x and for each x we evaluate MVE, ST and ZGR once, which is sufficient to compute their exact bias and variance. Since GRMC is stochastic, we evaluate it 1000 times (each time it draws M Gumbel samples conditioned on x and averages). Given the resulting estimates in 1000 trials we compute the empirical squared bias and variance using the unbiased estimators from Appendix D. Hence, the estimated squared bias can be negative, as a result plots are truncated to the positive range.

E.2 Variational Auto-Encoder

We used the MNIST dataset LeCun and Cortes (2010) and trained a discrete-latent VAE: The encoder architecture is:

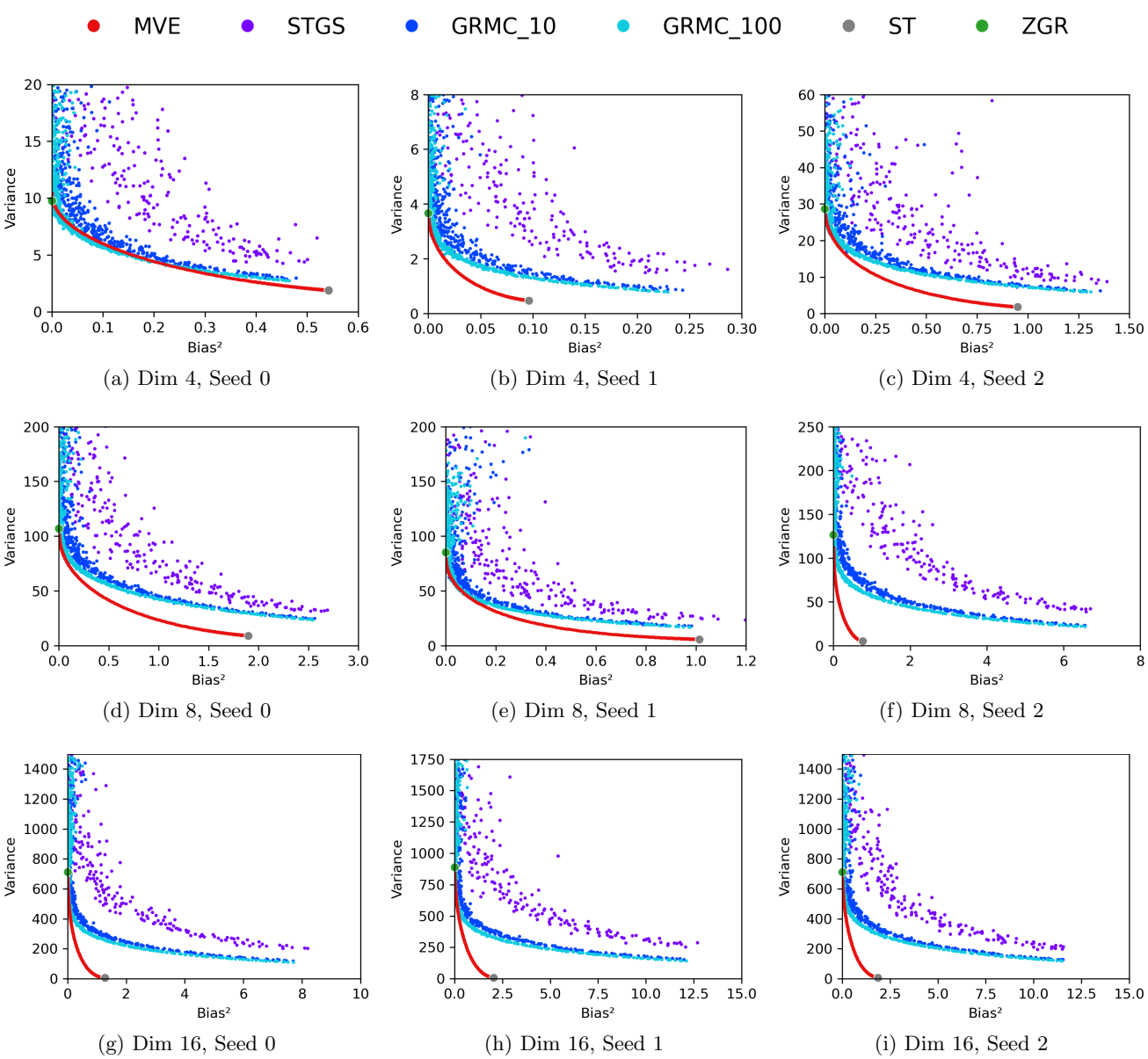
$$\text{Input}(784) \to \text{Linear}(512) \to \text{LeakyReLU}(0.2) \to \text{Linear}(256) \to \text{LeakyReLU}(0.2) \to \text{Linear}(n \times K).$$

The probabilistic model of categorical variables is the softmax of the encoder output. Accordingly, the encoder network outputs $K \times V$ logits. The decoder architecture is symmetric, mirroring the encoder. It inputs categorical variables in the one-hot encoding (a tensor of size $(n \times K)$) and outputs means of a Gaussian distribution. The decoder noise σ was a learned constant initialized as 1. We use a uniform prior $p(z)$ over discrete latent variables and closed form KL divergence. All models are trained using a batch-size of 200, AdamW, and a cosine annealing scheduler for 500 epochs. The latent space is integer or categorical with n independent variables, each having K categories. The latent-spaces, described in Table E.1 are chosen such that the first layer of the decoder has a constant number of parameters for each embedding. This ensures that the encoder's capacity remains constant across different latent space designs. However, the amount of information that can be encoded in the latent space varies significantly with n and K. For instance, the 128×4 configuration amounts to 256 bits of information while the 8×64 configuration amounts to $64^8 = 2^{48}$ states, *i.e.*, 48 bits. Therefore achievable negative ELBO (NELBO) loss can be expected to increase with the smaller latent space capacity but it is bounded also by the decoder capacity which stays constant.

The exponential schedule used for the bias-variance annealing in GRMC is

$$t(k) = t_{\text{start}} \left(\frac{t_{\text{end}}}{t_{\text{start}}}\right)^{\frac{k}{\text{max\_epochs}}}, \tag{165}$$

where k is the current epoch, max_epochs is the total number of epochs, $t_{\text{start}} = 1$ and $t_{\text{end}} = 0.1$. For MVE, we use the quadratic bias penalty coefficient τ with the exponential schedule, where $\tau_{\text{start}} = 1$ and $\tau_{\text{end}} = 1e^6$. We did not attempt to optimize hyperparameters of these schedules such as starting and final temperatures.



(a) Dim 4, Seed 0 (b) Dim 4, Seed 1 (c) Dim 4, Seed 2

(d) Dim 8, Seed 0 (e) Dim 8, Seed 1 (f) Dim 8, Seed 2

(g) Dim 16, Seed 0 (h) Dim 16, Seed 1 (i) Dim 16, Seed 2

Figure E.1: Bias^2-variance plots for different quadratic loss functions for categorical dimensions K =4, 8, and 16.

E.2.1 Integer VAE

In the Integer VAE variant, the encoder parameterizes a discrete categorical distribution over K classes for each latent dimension. To interface with the decoder, we project the sampled one-hot category vectors into a single scalar value using the embedding mapping $\Phi = (0, 1, \ldots, K-1)/(K-1)$. This transformation embeds the distinct categories as linearly spaced points along the interval $[0, 1]$. Consequently, as the number of categories K grows large ($K \gg 1$), the spacing between adjacent points approaches zero. This allows the discrete categorical latent space to smoothly approximate a bounded, continuous latent representation. The rest of the network is identical to the VAE with one-hot embedding latent space.

E.2.2 Greedy Search for Optimal Schedules

The parameterization of the temperature schedule for each gradient estimator can significantly impact its performance. Furthermore, the design of these schedules differs depending on the estimator being used and is entirely

Table E.1: VAE experiment with integer and categorical latents, same as in Table 2, but including per-experiment standard deviations over 5 runs (not std of the reported mean, which would be $1/\sqrt{5}$ factor less). Superscripts indicate the optimal learning rate found: ^{a} learning rate $= 10^{-3}$; ^{b} learning rate $= 3.2 \times 10^{-3}$.

Integer Latents

Latents $V \times K$	128×4	128×16	128×64	128×256	128×1024
ST	112.87 ± 0.06^b	113.71 ± 0.07^b	113.79 ± 0.13^b	113.80 ± 0.06^b	113.82 ± 0.06^b
GRMC-20 Exp	96.77 ± 0.18^a	100.06 ± 0.32^b	100.19 ± 0.16^b	103.12 ± 0.71^b	110.10 ± 0.86^b
ZGR	95.97 ± 0.16^b	94.52 ± 0.13^a	94.84 ± 0.12^a	95.72 ± 0.17^a	97.94 ± 0.41^a
ReinMax ($t = 1.2$)	96.27 ± 0.05^b	94.77 ± 0.14^a	95.04 ± 0.12^a	96.03 ± 0.12^a	98.15 ± 0.11^b
ReinMax ($t = 1.4$)	96.99 ± 0.13^b	95.33 ± 0.19^a	95.66 ± 0.21^a	96.90 ± 0.43^a	99.70 ± 0.60^b
1D MVE Exp	95.82 ± 0.09^b	93.37 ± 0.05^b	$\mathbf{93.24 \pm 0.08}^b$	$\mathbf{93.24 \pm 0.21}^b$	$\mathbf{93.23 \pm 0.10}^b$
1D MVE	$\mathbf{95.71 \pm 0.12}^b$	$\mathbf{93.12 \pm 0.09}^b$	93.30 ± 0.17^b	93.33 ± 0.05^b	93.33 ± 0.15^b

One-hot Categorical Latents

Latents $V \times K$	128×4	32×16	8×64
ST	112.47 ± 0.11^b	112.75 ± 0.10^b	111.40 ± 0.17^b
GRMC-20 Exp	96.31 ± 0.22^a	95.76 ± 0.16^b	101.49 ± 0.29^b
ZGR	95.96 ± 0.18^a	94.36 ± 0.24^a	98.29 ± 0.64^a
ReinMax ($t = 1.2$)	96.22 ± 0.26^b	$\mathbf{93.50 \pm 0.17}^b$	93.73 ± 0.20^b
ReinMax ($t = 1.4$)	96.60 ± 0.31^b	93.89 ± 0.26^a	93.72 ± 0.17^b
MVE Exp	$\mathbf{95.37 \pm 0.21}^a$	94.60 ± 0.23^b	$\mathbf{93.64 \pm 0.35}^b$

heuristic. This makes a direct comparison of the underlying estimators challenging; it is difficult to isolate whether improved model performance results from the inherent properties of the estimator itself, or simply from a superior, hand-tuned temperature schedule.

To ensure a rigorous and fair comparison, we circumvent this problem by approximating an optimal temperature schedule through a greedy search algorithm. Specifically, we divide the training process into discrete evaluation windows of 25 epochs. At the beginning of each cycle, we evaluate a grid of candidate temperatures. After 25 epochs, we select the temperature that achieves the highest ELBO value. The model's global state, optimizer, and scheduler are then updated with the weights from this winning run, the optimal temperature is recorded, and the process repeats for the next window. While a locally optimal temperature over a 25-epoch horizon does not strictly guarantee a globally optimal schedule over the entire training duration, computing the true global optimum is computationally intractable. Therefore, this greedy algorithm serves as a robust and reasonable approximation to normalize temperature effects across different estimators.

Training Details and Hyperparameters

For this experiment, we use the architecture, batch size, optimizer, epoch count, and learning rate schedule detailed at the beginning of this section. We set the learning rate to the value that yielded the best performance in the baseline VAE experiment using heuristic temperature schedules (see Table 2). To optimize the search space for each specific estimator during the greedy selection, candidate temperatures were sampled from estimator-specific grids over two "zoom levels" to fine-tune the selection. Specifically, candidates for MVE were drawn from a geometric progression ranging from 0.1 to $1,000,000$, while candidates for Gumbel-Rao were drawn from a linear space ranging from 0.1 to 1.0.

E.2.3 Full Profiling over Learning Rates and Temperatures

The greedy search approach cannot be used to fairly compare ReinMax against other estimators as it relies on a fixed temperature parameter. We note that the role of temperature differs fundamentally between the two methods: in MVE, it is used to reduce variance, whereas in ReinMax, it serves to smooth the surrogate loss surface during the backward pass. To ensure a rigorous and equitable comparison, we conducted a comprehensive grid search over both the temperature and the learning rate. Evaluating these parameters jointly is crucial, as

their effects on training dynamics are highly coupled.

Experimental Setup and Hyperparameters

To conduct this comparison, we evaluated 900 unique hyperparameter configurations for each estimator, resulting in $1,800$ total training runs. The network was trained for 50 epochs with a latent space consisting of 128 categorical variables each with 4 classes.

The grid search evaluated 30 learning rates and 30 temperatures for each estimator. Learning rates were sampled from a geometric progression between 10^{-4} and 10^{-2}. The evaluated temperatures were linearly spaced between 1.01 and 1.30 for ReinMax, and reciprocally spaced between 5 and 150 for MVE. These specific temperature and learning rate ranges were chosen to provide a full description of the qualitative behavior of each estimator (see Fig. E.2).

The resulting plot demonstrates that ReinMax and MVE achieve highly comparable overall performance, with their contour surfaces exhibiting a similar shape. ReinMax does hold a slight edge on this specific task, achieving a minimum negative ELBO that is 0.25 lower than MVE. However, given this narrow margin, along with the resolution and asymmetry of the sample grids, the results do not definitively establish one estimator as strictly superior. It is important to note that temperature serves a very different role in each method, and the underlying structures of these estimator families are qualitatively different. As a result, their relative performance may diverge when applied to different tasks or more complex loss surfaces.

E.3 Running Time

VAE In VAE the the cost of the decoder forward-backward is rather high and therefore cubic complexity for handling the gradient thorough latents is not necessarily a bottleneck for moderate K. For the latent space of 32 16-way categorical variables (where the complexity for our method would be the highest) and the encoder-decoder architecture used in the main paper, we have measured the following times on the Nvidia RTX4070 GPU:

- ST: 14 s/epoch
- ZGR: 14 s/epoch
- Gumbel-Rao (20 inner MC samples, computed in parallel): 15 s/epoch
- MVE: 16 s/epoch

Quantization In quantization, the speed of applying MVE in GPU can be approximately the same as that of straight-through, because it has linear complexity and the training is typically bounded by the memory bandwidth or the quadratic complexity of linear / convolution layers. Currently, as we observe training performance improvement of MVE over ZGR but not over ST, this has no foreseeable practical relevance, nevertheless, the timings for the experiment in Fig. 4 are as follows (wall-clock forward-backward time on A100 GPU):

- ST: 30 s/epoch
- ZGR: 63 s/epoch
- MVE: 60 s/epoch

In this case we are using torch.compile on the gradient estimator, but the implementations can be likely improved further.

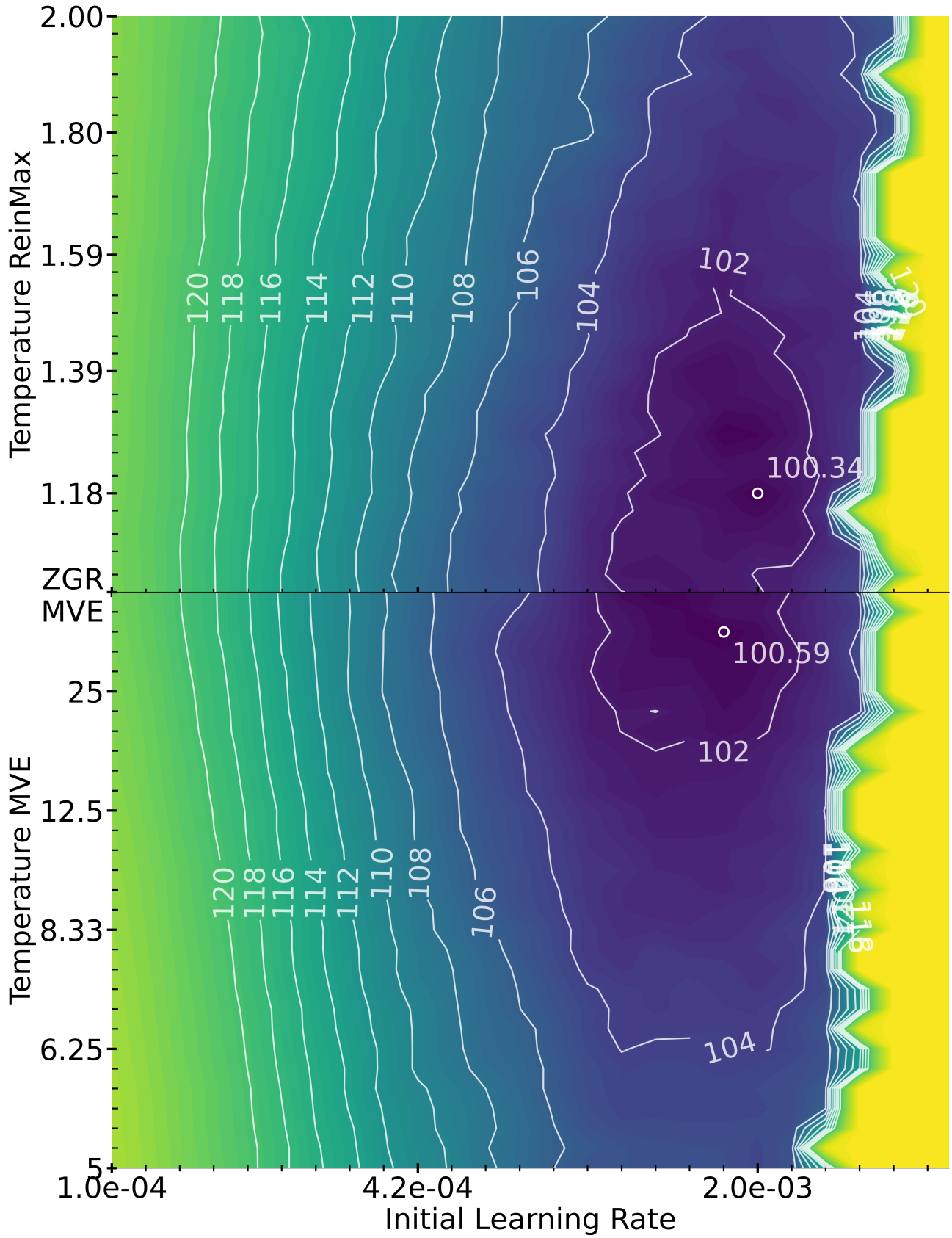


Figure E.2: Parametric ReinMax vs MVE: the plot shows the final NELBO for different temperatures and learning rates of ReinMax and MVE.