WHEN DISENTANGLEMENT IS NOT ENOUGH: IDENTIFIABILITY AND SEMANTIC PRIVACY IN LEARNED COMMUNICATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Advances in representation learning allow for new and previously unfeasible approaches for efficient and secure data transmission over noisy channels. A wireless communication system based on separate source coding, i.e., compression, encryption, and channel coding, i.e., error correcting coding, is proven to be asymptotically optimal and is currently widely used in modern secure wireless networks. Joint source-channel coding is an alternative with a range of advantages over the separation based scheme, including a better communication performance in short data blocklengths and graceful performance degradation with worsening channel quality. Although the integration of encryption, including feature-selective encryption, into the joint source-channel coding is addressed in the literature, it is not studied in the context of capabilities of the existing representation learning models. In this work, we first prove that zero information leakage about the private semantic concept from public latent components is generally not achievable in unsupervised representation learning. We then show that weakly supervised disentangled representation learning models based on variational autoencoders can overcome this limitation and can be used as a base model for semantic factor disentanglement in a joint source-channel-encryption coding.

1 INTRODUCTION

One can achieve a substantially better use of wireless communication resources, if considers design of the compression and error correction algorithms jointly while communicating only the semantically relevant information to the receiver. This is clearly demonstrated with the joint source-channel coding (JSCC) methods, both theoretically and algorithmically (Kostina & Verdú, 2012; Choi et al., 2019; Bourtsoulatze et al., 2019). Opposite to JSCC is the separation of source coding (compression), channel coding (error correction coding) where encryption (if security is of concern) is commonly embedded in between source and channel coding as a stand-alone algorithm. In the JSCC with encryption, the major problem is the impact of stand-alone encryption algorithm on the JSCC performance. The JSCC can exploit source and channel statistics to achieve optimal performance. For example, the JSCC encoder can map similar source samples to nearby points in the channel input space such that, despite the channel noise, the source samples can be recovered with minimal possible distortion given the communication rate constraint. Meanwhile, conventional encryption algorithms may alter the structure or distribution of the transmitted representation and may not be robust to channel noise, reducing the effectiveness of the JSCC.

A method to integrate encryption into JSCC, keeping it effective, is proposed by Tung & Gündüz (2023), where the authors consider an autoencoder with non-trainable noisy channel layer. Tung & Gündüz (2023) encrypt the whole latent layer, i.e., it is not feature-selective, using homomorphic encryption based on learning with errors (LWE) problem (Regev, 2024). An advantage of model in Tung & Gündüz (2023) is that it is secure against chosen-plaintext attacks. A more sophisticated joint source-channel-encryption coding (JSCEC) model is proposed by Sun et al. (2024), where only a group of private features is protected while other group remains public. Encryption of only private semantic features compared to encryption of the entire latent space may allow for a less use of communication resources, e.g., the cryptographic key resource and the computation resource. The encryption stage in (Sun et al., 2024) is a learned network. Specifically, a neural network

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

that takes the private subcodeword together with a password and returns the protected block, and a second learned network at the receiver deciphers it. The model is then trained against a modelled eavesdropper and the security in model of (Sun et al., 2024) is empirical rather than information-theoretic.

The JSCC relies on representation learning, and feature-selective JSCEC imposes an additional requirement: the representation learning algorithm must learn distinct private and public semantic factors. There are a variety of variational autoencoder (VAE)-based models that are capable of learning disentangled data representations in unsupervised settings (Higgins et al., 2017; Chen et al., 2019). It is known, however, that a perfect disentanglement is generally not possible without inductive bias (weak supervision) (Locatello et al., 2019). As an example, the weakly supervised model Ada-GVAE provides substantially better performance in terms of disentanglement quality compared to unsupervised ones (Locatello et al., 2020).

In this paper, we study how existing representation learning models can enable JSCEC with feature-selective encryption, i.e., encryption of the private semantic factors only. We focus on the VAE (Kingma & Welling, 2022) architecture, and we prove that unsupervised models, where one is required to split latent layer into public and private groups, cannot generally guarantee zero information leakage about a private semantic concept. We then show that weak supervision can enable well-performing JSCEC. We also consider base models with a Lie Group latent layer (Higgins et al., 2018; Quessard et al., 2020; Zhu et al., 2021), whose private part can be encrypted directly as a Group element with a smaller key.

The main finding of our work is that feature-selective secrecy in learned JSCC is a representation learning problem. Unsupervised disentanglement cannot guarantee that a designated private factor is absent from the public part of the representation. A latent layer with independent components, which is the target of unsupervised disentanglement, can exhibit large amounts of leakage about the private factor. Weak supervision with pairs that share public factors resolves this ambiguity. The contributions of this work are as follows:

1. We show that the leakage of a private factor into the public part of the latent layer is not identifiable from the data distribution, i.e., unsupervised learning cannot guarantee zero information leakage (Theorem 1). With pairs that share public factors the leakage is zero in the limit, and when private and public factors are dependent it reaches its lowest value, i.e., the information that public factors themselves carry about private ones (Theorem 2 and Proposition 1).

2. We propose a feature-selective JSCEC, where private channel symbols are encrypted by a keyed Haar random rotation, and we show that the leakage to the eavesdropper is bounded by the leakage of the base model (Proposition 2), i.e., the secrecy of JSCEC reduces to the quality of the representation.

3. We show on four datasets that with Ada-GVAE as a base model the eavesdropper's advantage over random guess on private factors is low on three datasets, also without channel noise, and that the remaining leakage follows the disentanglement quality of the base model.

4. We show that when the private part takes values in a compact group, e.g., in a base model with a Lie group latent layer, it can be encrypted by a Haar random group element without the constant power requirement and with a smaller key.

2 Joint Source-Channel-Encryption Coding

In this section we introduce the problem of JSCEC. We briefly outline general model with explanations of roles for each JSCEC stage, i.e., source coding, channel coding and encryption. We then present a method to implement JSCEC using VAE-based architecture. We also present a limitation of unsupervised representation learning models for JSCEC. We then discuss weakly supervised disentanglement Ada-GVAE model and Lie Group-based disentanglement model in the context of JSCEC.

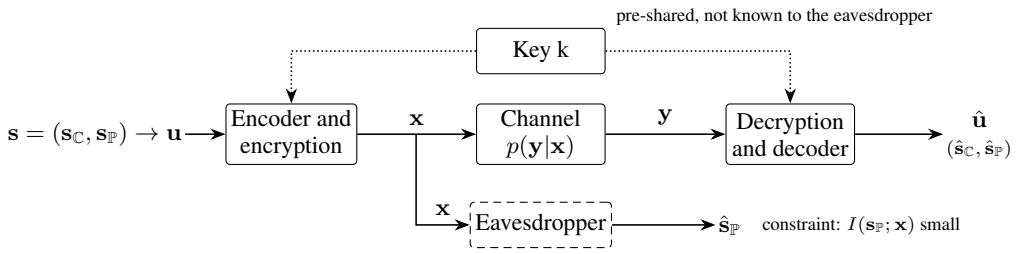


Figure 1: General model of JSCEC with feature-selective encryption.

2.1 General Model

We consider a wireless communication model shown in Fig. 1. Source samples $\mathbf{u} \in \mathbb{U}^k$, accompanied with ground-truth semantic factors $\mathbf{s} = (\mathbf{s}_\mathbb{C}, \mathbf{s}_\mathbb{P}) \in \mathbb{S}^d$, are encoded and encrypted to produce the channel input $\mathbf{x} \in \mathbb{R}^n$ in such a way that private semantic factors $\mathbf{s}_\mathbb{P}$ are to remain secret while semantic factors $\mathbf{s}_\mathbb{C}$ may be public. The channel input is then sent over a noisy channel defined by a conditional probability distribution $p(\mathbf{y}|\mathbf{x})$. The transmitter maps $\mathbf{u}$ into n uses of channel, i.e., $\mathbf{x}$, given transmitted power constraint $\frac{1}{n}\mathbb{E}[||\mathbf{x}||_2^2] \leq P$. The legitimate receiver, using the pre-shared secret key k, decodes and decrypts $\mathbf{u}$ and $(\mathbf{s}_\mathbb{C}, \mathbf{s}_\mathbb{P})$ from the channel output $\mathbf{y}$. The eavesdropper, either directly or through its channel, obtains $\mathbf{x}$ and attempts to decode $\mathbf{u}$ and $(\mathbf{s}_\mathbb{C}, \mathbf{s}_\mathbb{P})$ without the decryption key. One of the objectives of the transmitter is to protect semantic factors, $\mathbf{s}_\mathbb{P}$, from being revealed. Other objectives are to provide some level of reconstruction quality at the receiver and to guarantee some error correction capabilities against the channel noise.

2.1.1 Source Coding

The general goal of the source coding stage is to represent source data with a minimal message size, given that it should be then decoded with some level of fidelity at the receiver. Information theory allows one to characterise fundamental limits of compression in terms of data rate and data fidelity, i.e., rate-distortion function (El Gamal & Kim, 2011), including cases in which data source is not independent and identically distributed (i.i.d) (Han, 2003).

Common examples of source coding algorithms used in separation-based communication systems and beyond are JPEG, PNG, BPG for images, MP3, FLAC for audio, HEVC for video, etc (Wallace, 1992; Sullivan et al., 2012). In machine learning (ML)-based JSCC, the source coding stage is closely related to representation learning and usually is implemented as a deep neural network of the autoencoder type (Bourtsoulatze et al., 2019).

Since that model should provide secrecy for semantic private factors, the source coding may encode data into two blocks $\mathbf{z} = (\mathbf{z}_\mathbb{C}, \mathbf{z}_\mathbb{P}) \in \mathbb{R}^d$, in a way to minimise the amount of information about private factors contained in the public block. This constitutes a representation learning problem.

2.1.2 Channel Model and Channel Coding

In wireless communications, the channel allows one to model noise introduced by wireless medium to the transmitted data. The channel coding follows the source coding stage with the main goal to provide robustness of encoded information to the channel noise, i.e., enable error correction at the receiver.

In the case of feature-selective JSCEC, where encryption comes after channel coding, the channel coding should map the two groups of the latent layer into two groups of channel symbols and should not mix these groups, as otherwise the public channel symbols may carry information about the private group.

2.1.3 Encryption

The source coding and the channel coding alone can enable secrecy in wireless communication via privacy funnel (Makhdoumi et al., 2014) and wiretap coding (Wyner, 1975), respectively. How-

ever, in this work we focus our study on shared-key encryption to be the main enabling security mechanism. We also focus on the case when only some semantic features of data are encrypted.

2.2 Proposed Method

To implement the general model presented in Subsection 2.1, we propose the following structure. We use the disentanglement VAE-like base model, followed by channel coding stage, implemented by a multilayer perceptron (MLP), and feature-selective encryption, implemented by a matrix transform.

We consider a VAE setup (Kingma & Welling, 2022). We define a concept p, i.e., a word or a short phrase, that induces a partition $\{1, ..., d\} = \mathbb{C} \sqcup \mathbb{P}$ of latent layer into public (common) and private parts $\mathbf{z} = (\mathbf{z}_\mathbb{C}, \mathbf{z}_\mathbb{P})$. In general, the concept p can define the public or the private part, in this paper we assume, without loss of generality, that p sets the private part.

2.2.1 Limitation of Unsupervised Representation Learning

We first study whether an unsupervised representation can theoretically guarantee that a designated private semantic concept is absent from the public part of the representation. This problem is related to semantic-features disentanglement, however, is more subtle, i.e., a representation may be fully disentangled but still assign private semantic factors to public latent components and thus leak the private concept. Consider the following setup. Define true semantic factors as $\mathbf{s} = (\mathbf{s}_\mathbb{P}, \mathbf{s}_\mathbb{C}) \in \mathbb{S} \subseteq \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, $d = d_1 + d_2$ and $d_1, d_2 \geq 1$. Let $p(\mathbf{s}) = \prod_{i=1}^{d} p(\mathrm{s}_i)$, where each $p(\mathrm{s}_i)$ is a continuous density with support on an open interval. In particular $\mathbf{s}_\mathbb{P} \perp \mathbf{s}_\mathbb{C}$. Let $\mathbf{u} = g^*(\mathbf{s})$, where a true generating function $g^* : \mathbb{S} \to \mathbb{U}$ is a smooth bijection with a smooth inverse. Also let an encoder $\mathbf{z} = (\mathbf{z}_\mathbb{P}, \mathbf{z}_\mathbb{C}) = e(\mathbf{u})$, and we measure the information leakage using mutual information $I(\mathbf{s}_\mathbb{P}; \mathbf{z}_\mathbb{C})$.

Definition 1 *A learned model $(\hat{p}, \hat{g})$ is admissible if $\hat{p}$ is a factorised density on $\hat{\mathcal{Z}} \subseteq \mathbb{R}^d$, $\hat{g} : \hat{\mathcal{Z}} \to \mathbb{U}$ is a smooth bijection, and $\hat{g}(\mathbf{z}) \overset{d}{=} \mathbf{u}$ for $\mathbf{z} \sim \hat{p}$, i.e., decoded samples from $\hat{p}$ produce the data distribution. Its encoder is $e = \hat{g}^{-1}$.*

We develop from results of (Locatello et al., 2019) and (Locatello et al., 2020) and present the following two theorems and one proposition about representation learning models. Proofs of Theorems 1 and 2 and Proposition 1 are presented in Appendix A.1.

Theorem 1 *Given the above setting, there is a set of admissible models $\{(p(\mathbf{s}), \hat{g}_\theta)\}_{\theta \in [0, \pi/2]}$[1] with latent layer $\mathbf{z}^\theta = \hat{g}_\theta^{-1}(\mathbf{u})$ such that every admissible model has the same $p(\mathbf{s})$ and induces the same $p(\mathbf{u})$. Also $\mathbf{z}^\theta \overset{d}{=} \mathbf{s}$ for every θ, in particular $I(\mathbf{z}^\theta_\mathbb{P}; \mathbf{z}^\theta_\mathbb{C}) = 0$, and $I(\mathbf{s}_\mathbb{P}; \mathbf{z}^\theta_\mathbb{C}) = -\log\cos(\theta)$ for $\theta \in [0, \pi/2)$, so the information leakage takes every value in $[0, \infty)$. For example, at $\theta \to \pi/2$ one public latent component is an invertible function of one private factor, i.e., the private factor is fully exposed in the public part.*

Consequently, $I(\mathbf{s}_\mathbb{P}; \mathbf{z}_\mathbb{C})$ is not identifiable from the data distribution $p(\mathbf{u})$ alone. However, for weakly supervised scenario we establish the following result.

Theorem 2 *Given the above setting, suppose pair setup $\mathbf{u} = g^*(\mathbf{s}_\mathbb{P}, \mathbf{s}_\mathbb{C})$ and $\mathbf{u}' = g^*(\mathbf{s}'_\mathbb{P}, \mathbf{s}_\mathbb{C})$, where $\mathbf{s}'_\mathbb{P} \sim p(\mathbf{s}_\mathbb{P})$ is independent of $(\mathbf{s}_\mathbb{P}, \mathbf{s}_\mathbb{C})$. An admissible pair model $(\hat{p}, \hat{g}, \mathbb{T})$ consists of a factorised density $\hat{p}$ with product support $\hat{\mathcal{Z}} = \hat{\mathcal{Z}}_\mathbb{T} \times \hat{\mathcal{Z}}_{\mathbb{T}^c}$, a smooth bijection $\hat{g} : \hat{\mathcal{Z}} \to \mathbb{U}$ and an index set $\mathbb{T} \subset \{1, ..., d\}$ with $|\mathbb{T}| = d_2$. It generates $\mathbf{u} = \hat{g}(\mathbf{z})$ and $\mathbf{u}' = \hat{g}(\mathbf{z}')$, where $\mathbf{z}'_\mathbb{T} = \mathbf{z}_T$ and $\mathbf{z}'_{\mathbb{T}^c} \sim \hat{p}(\mathbf{z}_{T^c})$. If such a model reproduces $p(\mathbf{u}, \mathbf{u}')$ and we set $\mathbf{z}_\mathbb{C} = \mathbf{z}_T$ and $\mathbf{z}_\mathbb{P} = \mathbf{z}_{\mathbb{T}^c}$ with $\mathbf{z} = \hat{g}^{-1}(\mathbf{u})$, then $\mathbf{z}_\mathbb{C} = f(\mathbf{s}_\mathbb{C})$ for an injective function f, and $I(\mathbf{s}_\mathbb{P}; \mathbf{z}_\mathbb{C}) = 0$, $I(\mathbf{s}_\mathbb{C}; \mathbf{z}_\mathbb{P}) = 0$.*

This means that the public part of the latent layer keeps all information about the public factors and has no information about private factors. For dependent factors we have the following result.

Proposition 1 *Consider the setting of Theorem 2 without the independence of $\mathbf{s}_\mathbb{P}$ and $\mathbf{s}_\mathbb{C}$. Let pairs be generated with $\mathbf{s}'_\mathbb{P} \sim p(\mathbf{s}_\mathbb{P}|\mathbf{s}_\mathbb{C})$, and let the pair model generate $\mathbf{z}'_{T^c} \sim \hat{p}(\mathbf{z}_{\mathbb{T}^c}|\mathbf{z}_\mathbb{T})$, where $\hat{p}$ is not necessarily factorised. Then any encoder whose public part determines public factors, i.e.,*

[1] θ represents rotation of the Gaussian vector. See Appendix A.1 for details.

$\mathbf{s}_{\mathbb{C}} = f(\mathbf{z}_{\mathbb{C}})$ *for some function* f*, satisfies* $I(\mathbf{s}_{\mathbb{P}}; \mathbf{z}_{\mathbb{C}}) \geq I(\mathbf{s}_{\mathbb{P}}; \mathbf{s}_{\mathbb{C}})$*, and any admissible pair model that reproduces* $p(\mathbf{u}, \mathbf{u}')$ *achieves this bound with equality.*

Therefore, $I(\mathbf{s}_{\mathbb{P}}; \mathbf{s}_{\mathbb{C}})$ is the lowest leakage without discarding part of public factors. Theorem 2 and Proposition 1 are limits, i.e., they assume infinite data and exact matching of the data distribution. Ada-GVAE (Locatello et al., 2020) is trained with pairs that share all but one random factor, with which factors are identifiable only up to a permutation, so we assign latent components to the private part with labelled samples.

2.2.2 Base Disentanglement Models

In our model we consider two models as a base model for disentanglement, weakly supervised Ada-GVAE (Locatello et al., 2020) and Lie-VAE (Zhu et al., 2021). The weak-supervision in Ada-GVAE consists of presenting a pair of source samples which vary in at least one semantic factor. In Lie-VAE, the decoder takes the group element

$$\mathbf{g} = \exp(\frac{1}{\sqrt{d}} \sum_{j=1}^{d} \mathbf{z}_j \boldsymbol{A}_j), \tag{1}$$

where $\mathbf{z}$ are coordinates in a Lie algebra and $\boldsymbol{A}_j \in \mathbb{R}^{d_g \times d_g}$ are learned basis matrices. With skew-symmetric $\boldsymbol{A}_j$ we have $\mathbf{g} \in SO(d_g)$, i.e., the group is compact, and we refer to this model as Lie-VAE, and to the model with unconstrained $\boldsymbol{A}_j$ and without the factor $\frac{1}{\sqrt{d}}$ as non-compact Lie-VAE. The objective of Lie-VAE is the VAE objective with two additional penalties, which push the latent basis towards the Group structure (Zhu et al., 2021).

2.2.3 Channel Model and Channel Coding

We introduce the channel coding as the three-layer MLP in between latent layer and channel to make the latent representation more robust to the channel noise. The channel coding maps each latent block into power normalised channel symbols of a larger size. Specifically, $\mathbf{z}_{\mathbb{C}}$ and $\mathbf{z}_{\mathbb{P}}$ are encoded by two separate MLPs into $\mathbf{x}_{\mathbb{C}} \in \mathbb{R}^{n-m}$ and $\mathbf{x}_{\mathbb{P}} \in \mathbb{R}^{m}$, and each block is normalised in every frame, i.e., $||\mathbf{x}_{\mathbb{C}}||_2^2 = n - m$ and $||\mathbf{x}_{\mathbb{P}}||_2^2 = m$, which is required by the encryption introduced in Subsection 2.2.4.

We model the channel noise as additive white Gaussian noise (AWGN):

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n). \tag{2}$$

The signal-to-noise ratio (SNR) in dB is defined as,

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} \frac{\frac{1}{n}\mathbb{E}[||\mathbf{x}||_2^2]}{\sigma^2}. \tag{3}$$

To make the model to varying channel conditions, we use SNR conditioning for the channel coding MLP with feature-wise linear modulation (FiLM) (Perez et al., 2017).

2.2.4 Encryption

For encryption, we use an algorithm based on the matrix transform, where the transforming matrix is distributed according to the Haar probability measure (Meckes, 2019). For such algorithm it is easy to show a perfect information-theoretic security guarantee, i.e., zero information leakage. The only left information leakage about the private group is from the public one and it is due to limitations of a representation learning model.

The encryption we use is as follows. Let the channel input $\mathbf{x}_{\mathbb{P}}$ to take values from the sphere $\mathbb{S}^{m-1}(\sqrt{m}) \doteq \{v \in \mathbb{R}^m : \|v\|_2^2 = m\}$, i.e., have constant average power. Let $\mu_{O(m)}$ be a Haar probability measure on the orthogonal group $O(m)$. For every communication frame t and for a fixed encryption key k generate $\Omega(\mathrm{k}, t) \sim \mu_{O(m)}$ independently of t. Operationally, we obtain

$\Omega(\mathrm{k}, t)$ using the keyed pseudorandom function, where we use key k as a seed[2]. To encrypt, we have

$$\mathbf{c} = \Omega \mathbf{x}_{\mathbb{P}}. \tag{4}$$

We note that for every $\mathbf{x}_{\mathbb{P}} \in \mathbb{S}^{m-1}(\sqrt{m})$, we have $\Omega \mathbf{x}_{\mathbb{P}} \sim \text{Unif}(\mathbb{S}^{m-1}(\sqrt{m}))$ and $I(\mathbf{x}_{\mathbb{P}}; \mathbf{c}) = 0$ (see Appendix A.1). To decrypt noisy $\mathbf{c}$, we apply

$$\mathbf{y}_{\mathbb{P}} = \Omega^T(\mathbf{c} + \mathbf{n}), \tag{5}$$

where $\mathbf{n}$ is any perturbation added to the encrypted signal. In the case $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ we have $\Omega^T \mathbf{n} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, i.e., encryption does not change the noise distribution and its statistics.

One can generate Ω using QR factorization (complexity is $O(m^3)$) of matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ with elements distributed as $\mathcal{N}(0, 1)$. Specifically $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ and $\boldsymbol{Q}$ can be used as Ω with some corrections (Mezzadri, 2007). One can apply Ω to $\mathbf{x}_{\mathbb{P}}$ with $O(m^2)$ operations, without forming Ω using the algorithm presented in (Stewart, 1980).

Let the whole channel input be $\mathbf{x} = (\mathbf{x}_{\mathbb{C}}, \mathbf{c})$, where $\mathbf{c} = \Omega \mathbf{x}_{\mathbb{P}}$ is the encrypted private block and $\mathbf{x}_{\mathbb{C}}$ is the public block, which is not encrypted. Then all information leakage is through the public channel symbols.

Proposition 2 *Let $\mathbf{x}_{\mathbb{P}} \in \mathbb{S}^{m-1}(\sqrt{m})$, let $\Omega \sim \mu_{O(m)}$ be independent of $(\mathbf{s}, \mathbf{z}, \mathbf{x}_{\mathbb{C}}, \mathbf{x}_{\mathbb{P}})$, and let $\mathbf{x}_{\mathbb{C}}$ be a function of $\mathbf{z}_{\mathbb{C}}$ and the SNR. Then $I(\mathbf{s}_{\mathbb{P}}; \mathbf{x}) = I(\mathbf{s}_{\mathbb{P}}; \mathbf{x}_{\mathbb{C}}) \leq I(\mathbf{s}_{\mathbb{P}}; \mathbf{z}_{\mathbb{C}})$.*

The proof is presented in Appendix A.1. Consequently, the leakage of the JSCEC is bounded by the leakage of the base model, which by Theorem 1 cannot be guaranteed to be zero for an unsupervised base model, and by Theorem 2 can be zero, in the limit, if the model is trained with pairs that share public factors.

The condition of $\mathbf{x} \in \mathbb{S}^{m-1}(\sqrt{m})$ is necessary to have $I(\mathbf{x}; \mathbf{c}) = 0$. However, generally, if $\mathrm{r} = \frac{||\mathbf{x}||_2^2}{m}$ is not constant and is a discrete r.v., then $I(\mathbf{x}; \mathbf{c}) = H(\mathrm{r})$, as Ω does not change the norm of $\mathbf{x}$. The proof of this fact is presented in Appendix A.1. When the private part takes values in a compact group, one can drop the constant $||\mathbf{x}||_2^2$ requirement and still achieve zero information leakage. Consider Lie-VAE base model we introduce in Subsection 2.2.2 in which private factors are carried in the form of

$$\mathbf{g} = \exp\left(\frac{1}{\sqrt{d}} \sum_{j \in \mathbb{P}} \mathrm{z}_j \boldsymbol{A}_j\right) \in SO(d_g), \tag{6}$$

where $\boldsymbol{A}_j$ are skew-symmetric matrices. Then similarly to (4) and (5), we define encryption and decryption as

$$\mathbf{c} = \Omega \mathbf{g}, \quad \hat{\mathbf{g}} = \Omega^T(\mathbf{c} + \mathbf{n}), \tag{7}$$

where $\Omega \sim \mu_{SO(d_g)}$, i.e., from the Haar measure on $SO(d_g)$. As the Haar measure on a compact group is invariant, $\Omega \mathbf{g} \sim \mu_{SO(d_g)}$ for every $\mathbf{g} \in SO(d_g)$, and we have $I(\mathbf{c}; \mathbf{g}) = 0$ with a smaller key, i.e., $d_g(d_g - 1)/2$ parameters against $m(m-1)/2$ for $O(m)$ with $m = d_g^2$.

2.2.5 Eavesdropper Model and Secrecy Metric

We consider an eavesdropper that is able to intercept transmitted signal even without channel noise, i.e., $\mathbf{x}$. We assume that the eavesdropper knows the legitimate receiver decoder (neural network), but does not have decryption key, e.g., seed to pseudorandom function to generate Ω. The main goal of the eavesdropper is to decode set of private features $\mathbf{s}_{\mathbb{P}}$, that are represented with $\mathbf{z}_{\mathbb{P}}$ and may be partially presented in $\mathbf{z}_{\mathbb{C}}$. For eavesdropper's attack, we consider a family of estimators (probes) of each private factor from the decoded latent layer: random forest (RF), gradient boosting, MLP, k-nearest neighbours (kNN), and logistic regression, trained on labelled samples and tested on other samples. We then consider a maximum accuracy over this family, and we define secrecy metric to be this accuracy over random guess of private factors. For the legitimate receiver we use the same probes.

[2]This make size of key smaller, comparing to using whole matrix as a key. However, this weakens information-theoretic guarantee of zero encryption information leakage. We assume that information leakage from using keyed pseudorandom function is negligible comparing to imperfect representations.

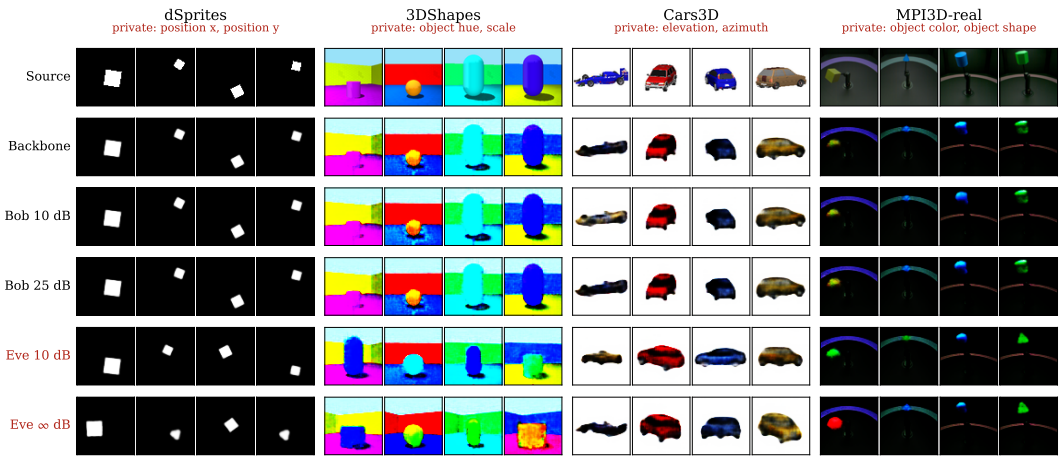


Figure 2: Reconstructions with Ada-GVAE base model: base model without channel (Backbone), legitimate receiver (Bob) at 10 and 25 dB, and eavesdropper (Eve) at 10 dB and without channel noise (∞ dB).

Let the private factor s_k take values in a finite set $\mathbb{S}_k, k \in \mathbb{P}$, $|\mathbb{S}_k| > 1$. The probability of uniform random guess is $\rho_k = 1/|\mathbb{S}_k|$, let the p-th probe accuracy for k-th private factor is $\pi_{k,p}$. The secrecy for k-th private factor is then defined as

$$A_k = \max_{p \in \{\text{probes}\}} \max\left(0, \frac{\pi_{k,p} - \rho_k}{1 - \rho_k}\right) \in [0, 1]. \tag{8}$$

To relate our experiments to the theory one can measure $I(\mathbf{s}_{\mathbb{P}}; \mathbf{z}_{\mathbb{C}})$, which is the leakage in Theorems 1 and 2 and an upper bound on $I(\mathbf{s}_{\mathbb{P}}; \mathbf{x})$ by Proposition 2. However, estimation of mutual information from samples is challenging Poole et al. (2019).

3 Experiments

In our experiments, we consider Ada-GVAE and Lie-VAE as a base models for feature disentanglement and we use channel coding and encryption presented in Subsection 2.2.3 and in Subsection 2.2.4, respectively. We use $d = d_g = 10$, $n = 24$ and $m = 12$. The base model is trained first, then frozen, and the channel coding is trained through the frozen decoder. See Appendix A.2 for more details.

3.1 Datasets

We evaluate proposed JSCEC models on dSprites (Matthey et al., 2017), 3DShapes (Burgess & Kim, 2018), Cars3D (Reed et al., 2015), and MPI3D-real (Gondal et al., 2019) datasets. See top row of Fig. 2 for samples from each dataset. These datasets are of varying complexity and are commonly used as a benchmark for disentanglement quality of representation learning methods. The private factors are position x and y in dSprites, object hue and scale in 3DShapes, elevation and azimuth in Cars3D, and object colour and shape in MPI3D-real.

3.2 Results

We train our models on three different seeds. To assess models' performance we use one fidelity metric, i.e., PSNR, MIG disentanglement metric (Chen et al., 2019), and the secrecy metric defined in Subsection 2.2.5. During training the SNR is scheduled to the target range $[0, 25]$ dB and provided to the channel encoder and decoder as a conditioning input, so that a single trained model covers the range rather than a separate model being trained at each SNR operating point. All figures are shown only within the SNR target range $[0, 25]$ dB.

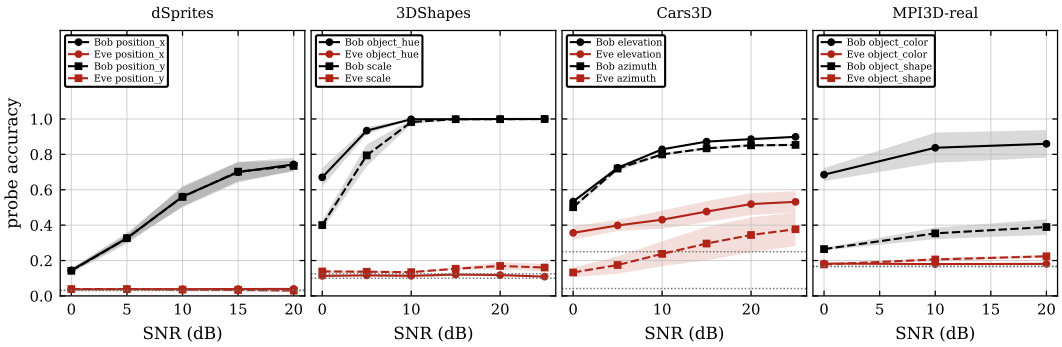


Figure 3: Accuracy of the best probe for each private factor vs SNR for the legitimate receiver (Bob, black curves) and the eavesdropper (Eve, red curves) with Ada-GVAE base model. Dotted lines show the probability of a random guess. We plot mean and standard deviation over three seeds.

Table 1: Secrecy metric A_k of the legitimate receiver (Bob) at 10 dB SNR and of the eavesdropper (Eve) tapping the channel input without noise while the transmitter operates at 10 dB SNR, mean $\pm$ standard deviation over three seeds. $A_k = 0$ is random guess and $A_k = 1$ is full recovery of the private factor.

		Ada-GVAE		Lie-VAE	
Dataset	Private factor	Bob	Eve	Bob	Eve
dSprites	position x	0.55 ± 0.06	0.007 ± 0.001	0.42 ± 0.01	0.023 ± 0.014
	position y	0.55 ± 0.06	0.010 ± 0.007	0.42 ± 0.02	0.018 ± 0.013
3DShapes	object hue	1.00 ± 0.00	0.029 ± 0.014	0.97 ± 0.02	0.084 ± 0.039
	scale	0.98 ± 0.02	0.058 ± 0.017	0.64 ± 0.21	0.179 ± 0.158
Cars3D	elevation	0.77 ± 0.01	0.411 ± 0.075	0.80 ± 0.01	0.359 ± 0.078
	azimuth	0.79 ± 0.01	0.369 ± 0.087	0.84 ± 0.02	0.240 ± 0.052
MPI3D-real	object colour	0.80 ± 0.10	0.024 ± 0.008	0.81 ± 0.12	0.088 ± 0.052
	object shape	0.22 ± 0.04	0.071 ± 0.004	0.19 ± 0.06	0.065 ± 0.034

We firstly report that the reconstruction quality measure with PSNR and the disentanglement quality measured with MIG metric is dependent on SNR. This is reflected in Fig. 4. Appendix A.3. The channel MLP and the channel noise reduce disentanglement[3] in MIG metric, this is reflected by the gap from dotted horizontal lines that represent MIG of the model without channel, channel coding and decryption. We also note that Lie-VAE shows worse disentanglement comparing to Ada-GVAE.

Secrecy against eavesdropping of private semantic factors is reflected in Fig. 3 and in Fig. 5 for Ada-GVAE and Lie-VAE base models[4], respectively, and Table 1 shows A_k at 10 dB SNR, where the eavesdropper observes noiseless channel input. With Ada-GVAE on dSprites, 3DShapes and MPI3D-real, the eavesdropper advantage on private factors stays close to the probability of a random guess at any SNR, while the legitimate receiver has reasonably high A_k, except object shape in MPI3D-real, which is poorly represented by the base disentanglement model. Public factors remain available to the eavesdropper, e.g., floor and wall hue in 3DShapes are recovered with high accuracy by both receivers, i.e., encryption is feature-selective, which is also visually seen in Fig. 2. Generally, the eavesdropper's advantage grows with SNR.

For Cars3D the information leakage about the private factors is substantial. The MIG of the base model is low for both Ada-GVAE and Lie-VAE, i.e., private factors are not localised and part of their information is left in the public part.

[3]One exception is 3DShapes dataset with Ada-GVAE, where MIG at high SNR is slightly above the one of the base model, although the difference is within the spread over seeds.

[4]Accuracy may go beyond the probability of the random guess due to noise and limited sample size provided to the probes

4 CONCLUSION

In this paper we study how representation learning models can enable feature-selective encryption in JSCEC scenario. We show that unsupervised learning cannot provide zero information leakage about the private concept. With Haar-distributed matrix rotation of the private channel symbols, the leakage to the eavesdropper is bounded by the leakage of the base representation learning model, which can be theoretically zero if the model is trained with pairs that share public factors and ground-truth public and private factors are independent. Our experiments show that with Ada-GVAE as a base model, the eavesdropper's performance is close to random guess of private factors of dSprites, 3DShapes and MPI3D-real at every SNR, while on Cars3D, where disentanglement is poor, the leakage is substantial.

In current work, we have used simple choice of base disentanglement model and encryption algorithm to illustrate the feasibility of JSCEC. In future work, one might consider JSCEC with more sophisticated representation learning methods as the base model and other encryption algorithms, which are cryptographically more developed. It is also reasonable to compare models' performance to other existing JSCC and JSCEC methods.

ACKNOWLEDGMENTS

This research is supported by the 5/6GIC, Institute for Communication Systems (ICS), University of Surrey.

REFERENCES

Eirina Bourtsoulatze, David Burth Kurka, and Deniz Gündüz. Deep joint source-channel coding for wireless image transmission. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4774–4778, 2019. doi: 10.1109/ICASSP.2019.8683463.

Chris Burgess and Hyunjik Kim. 3D shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders, 2019. URL https://arxiv.org/abs/1802.04942.

Kristy Choi, Kedar Tatwawadi, Aditya Grover, Tsachy Weissman, and Stefano Ermon. Neural joint source-channel coding, 2019. URL https://arxiv.org/abs/1811.07557.

Abbas El Gamal and Young-Han Kim. *Network Information Theory*. Cambridge University Press, 2011.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Te Sun Han. *Information-Spectrum Methods in Information Theory*, volume 50 of *Stochastic Modelling and Applied Probability*. Springer, Berlin, Heidelberg, 2003. ISBN 978-3-540-43581-5. doi: 10.1007/978-3-662-12066-8.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations, 2018. URL https://arxiv.org/abs/1812.02230.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Victoria Kostina and Sergio Verdú. Lossy joint source-channel coding in the finite blocklength regime. In *2012 IEEE International Symposium on Information Theory Proceedings*, pp. 1553–1557, 2012. doi: 10.1109/ISIT.2012.6283533.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations, 2019. URL https://arxiv.org/abs/1811.12359.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises, 2020. URL https://arxiv.org/abs/2002.02886.

Ali Makhdoumi, Salman Salamatian, Nadia Fawaz, and Muriel Medard. From the information bottleneck to the privacy funnel, 2014. URL https://arxiv.org/abs/1402.1774.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Elizabeth S. Meckes. *The Random Matrix Theory of the Classical Compact Groups*. Cambridge Tracts in Mathematics. Cambridge University Press, 2019.

Francesco Mezzadri. How to generate random matrices from the classical compact groups, 2007. URL https://arxiv.org/abs/math-ph/0609050.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017. URL https://arxiv.org/abs/1709.07871.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information, 2019. URL https://arxiv.org/abs/1905.06922.

Robin Quessard, Thomas D. Barrett, and William R. Clements. Learning group structure and disentangled representations of dynamical environments, 2020. URL https://arxiv.org/abs/2002.06991.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Oded Regev. On lattices, learning with errors, random linear codes, and cryptography, 2024. URL https://arxiv.org/abs/2401.03703.

G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis*, 17(3):403–409, 1980. ISSN 00361429. URL http://www.jstor.org/stable/2156882.

Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. doi: 10.1109/TCSVT.2012.2221191.

Lunan Sun, Yang Yang, Mingzhe Chen, and Caili Guo. Disentangled information bottleneck guided privacy-protective joint source and channel coding for image transmission. *IEEE Transactions on Communications*, 72(11):6992–7005, 2024. doi: 10.1109/TCOMM.2024.3406381.

Tze-Yang Tung and Deniz Gündüz. Deep joint source-channel and encryption coding: Secure semantic communications. In *ICC 2023 - IEEE International Conference on Communications*, pp. 5620–5625, 2023. doi: 10.1109/ICC45041.2023.10278612.

G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. doi: 10.1109/30.125072.

A. D. Wyner. The wire-tap channel. *The Bell System Technical Journal*, 54(8):1355–1387, 1975. doi: 10.1002/j.1538-7305.1975.tb02040.x.

Xinqi Zhu, Chang Xu, and Dacheng Tao. Commutative lie group vae for disentanglement learning, 2021. URL https://arxiv.org/abs/2106.03375.

A APPENDIX

AI USE STATEMENT

In this work, we used generative AI tools, specifically Anthropic's Claude Opus 5.0-5.5 and, to a lesser degree, OpenAI's ChatGPT 5.6 Luna, for implementing and testing model and running the experiments, for producing the figures, to formulate mathematical claims, assist in the writing of proofs, and for drafting and editing parts of the text. The theorems, the proofs, the results and the text were checked by the first author, who takes responsibility for the final content of this work.

A.1 PROOFS

Proof of Theorem 1. Fix a private index $i \leq d_1$ and a public index $j > d_1$. Let F apply the marginal cumulative distribution functions (CDF) of $p(\mathrm{s}_1), ..., p(\mathrm{s}_d)$ to each component, so that $F(\mathbf{s})$ is uniform on $(0,1)^d$, and let Φ apply the CDF of $\mathcal{N}(0,1)$ to each component. Then we have:

$$\mathbf{w} = \Phi^{-1}(F(\mathbf{s})) \sim \mathcal{N}(0, \boldsymbol{I}_d). \tag{9}$$

Let $\boldsymbol{R}_\theta$ be the rotation by angle θ in the plane of components (i,j), which does not change other components. As $\mathcal{N}(0, \boldsymbol{I}_d)$ is invariant to rotations, $\boldsymbol{R}_\theta \mathbf{w} \overset{d}{=} \mathbf{w}$. We define

$$f_\theta = F^{-1} \circ \Phi \circ \boldsymbol{R}_\theta \circ \Phi^{-1} \circ F, \tag{10}$$

which is a smooth bijection $\mathbb{S} \to \mathbb{S}$ with $f_\theta(\mathbf{s}) \overset{d}{=} \mathbf{s}$. Let $\hat{g}_\theta = g^* \circ f_\theta^{-1}$. Then for $\mathbf{z} \sim p(\mathbf{s})$ we have $\hat{g}_\theta(\mathbf{z}) = g^*(f_\theta^{-1}(\mathbf{z})) \overset{d}{=} g^*(\mathbf{s}) = \mathbf{u}$, so $(p(\mathbf{s}), \hat{g}_\theta)$ is admissible. Its latent layer is $\mathbf{z}^\theta = \hat{g}_\theta^{-1}(\mathbf{u}) = f_\theta(\mathbf{s})$, so $\mathbf{z}^\theta \overset{d}{=} \mathbf{s}$ and $I(\mathbf{z}^\theta_{\mathbb{P}}; \mathbf{z}^\theta_{\mathbb{C}}) = 0$.

Only components i and j are changed, and in the Gaussian coordinates the public component is $\mathrm{w}'_j = \sin(\theta)\mathrm{w}_i + \cos(\theta)\mathrm{w}_j$. Mutual information does not change under invertible transforms of each component, and all other components are independent of $(\mathrm{w}_i, \mathrm{w}_j)$ and of each other, so

$$I(\mathbf{s}_{\mathbb{P}}; \mathbf{z}^\theta_{\mathbb{C}}) = I(\mathrm{w}_i; \mathrm{w}'_j) = -\frac{1}{2}\log(1 - \sin^2(\theta)) = -\log\cos(\theta), \tag{11}$$

where we use that $(\mathrm{w}_i, \mathrm{w}'_j)$ is a Gaussian pair with correlation $\sin(\theta)$. At $\theta = \pi/2$ we have $\mathrm{w}'_j = \mathrm{w}_i$, so the public component $\mathrm{z}_j = F_j^{-1}(F_i(\mathrm{s}_i))$ is an invertible function of the private factor s_i.

Proof of Theorem 2. Matching $p(\mathbf{u}, \mathbf{u}')$ implies matching $p(\mathbf{u})$ and $p(\mathbf{u}'|\mathbf{u})$ for almost every $\mathbf{u}$. We compare supports of these conditional distributions. Let $\mathbb{S}^{(1)}$ be the support of $p(\mathbf{s}_{\mathbb{P}})$. Conditioning on $\mathbf{u}$ fixes $\mathbf{s} = (g^*)^{-1}(\mathbf{u})$ and hence $\mathbf{s}_{\mathbb{C}}$, while $\mathbf{s}'_{\mathbb{P}}$ takes values in $\mathbb{S}^{(1)}$, so we have

$$\operatorname{supp} p(\mathbf{u}'|\mathbf{u}) = g^*(\mathbb{S}^{(1)} \times \{\mathbf{s}_{\mathbb{C}}\}). \tag{12}$$

In the learned model, conditioning on $\mathbf{u}$ fixes $\mathbf{z} = \hat{g}^{-1}(\mathbf{u})$ and hence $\mathbf{z}_{\mathbb{T}}$, while $\mathbf{z}'_{\mathbb{T}^c}$ takes values in $\hat{\mathcal{Z}}_{\mathbb{T}^c}$, so similarly we have

$$\operatorname{supp} p(\mathbf{u}'|\mathbf{u}) = \hat{g}(\{\mathbf{z}_{\mathbb{T}}(\mathbf{u})\} \times \hat{\mathcal{Z}}_{\mathbb{T}^c}) = \{\mathbf{u}'' : \mathbf{z}_{\mathbb{T}}(\mathbf{u}'') = \mathbf{z}_{\mathbb{T}}(\mathbf{u})\}, \tag{13}$$

where we use that $\hat{g}$ is a bijection and $\hat{\mathcal{Z}}$ is a product set. The two supports are equal for almost every $\mathbf{u}$, so the level sets of the function $\mathbf{s} \mapsto \mathbf{z}_{\mathbb{T}}(g^*(\mathbf{s}))$ are the sets $\mathbb{S}^{(1)} \times \{\mathbf{s}_{\mathbb{C}}\}$. As $\mathbf{z}_T$ is constant on each of these sets, $\mathbf{z}_{\mathbb{T}} = f(\mathbf{s}_{\mathbb{C}})$ for some function f, and as $\mathbf{z}_T$ takes different values on different sets, f is injective. Then

$$I(\mathbf{s}_{\mathbb{P}}; \mathbf{z}_{\mathbb{T}}) = I(\mathbf{s}_{\mathbb{P}}; f(\mathbf{s}_{\mathbb{C}})) = I(\mathbf{s}_{\mathbb{P}}; \mathbf{s}_{\mathbb{C}}) = 0. \tag{14}$$

As $\hat{p}$ is factorised, $\mathbf{z}_{T^c} \perp \mathbf{z}_{\mathbb{T}}$, and $\mathbf{s}_{\mathbb{C}} = f^{-1}(\mathbf{z}_{\mathbb{T}})$ is a function of $\mathbf{z}_{\mathbb{T}}$, so $\mathbf{z}_{T^c} \perp \mathbf{s}_{\mathbb{C}}$ and $I(\mathbf{s}_{\mathbb{C}}; \mathbf{z}_{T^c}) = 0$.

Proof of Proposition 1. The first part follows from the data processing inequality (DPI), $I(\mathbf{s}_{\mathbb{P}}; \mathbf{z}_{\mathbb{C}}) \geq I(\mathbf{s}_{\mathbb{P}}; f(\mathbf{z}_{\mathbb{C}})) = I(\mathbf{s}_{\mathbb{P}}; \mathbf{s}_{\mathbb{C}})$. For the second part, the proof of Theorem 2 uses only supports of conditional distributions. In the true process the support of $p(\mathbf{u}'|\mathbf{u})$ is now $g^*(\operatorname{supp} p(\mathbf{s}_{\mathbb{P}}|\mathbf{s}_{\mathbb{C}}) \times \{\mathbf{s}_{\mathbb{C}}\})$, and in the model it is a subset of $\{\mathbf{u}'' : \mathbf{z}_{\mathbb{T}}(\mathbf{u}'') = \mathbf{z}_T(\mathbf{u})\}$ that depends on $\mathbf{u}$ only through $\mathbf{z}_T(\mathbf{u})$. So $\mathbf{z}_T$ is constant on each set $\operatorname{supp} p(\mathbf{s}_{\mathbb{P}}|\mathbf{s}_{\mathbb{C}}) \times \{\mathbf{s}_{\mathbb{C}}\}$, and two such sets with different $\mathbf{s}_{\mathbb{C}}$ cannot have the same value of $\mathbf{z}_T$, as then they would have the same conditional support. Therefore $\mathbf{z}_{\mathbb{T}} = f(\mathbf{s}_{\mathbb{C}})$ for an injective function f, and $I(\mathbf{s}_{\mathbb{P}}; \mathbf{z}_T) = I(\mathbf{s}_{\mathbb{P}}; \mathbf{s}_{\mathbb{C}})$.

Proof of Proposition 2. For every fixed $v \in \mathbb{S}^{m-1}(\sqrt{m})$ the distribution of Ωv is invariant to multiplication by any orthogonal matrix, as $\boldsymbol{U}\Omega \overset{d}{=} \Omega$ for every $\boldsymbol{U} \in O(m)$, and the only such probability distribution on the sphere is the uniform one, so $\Omega v \sim \text{Unif}(\mathbb{S}^{m-1}(\sqrt{m}))$. As Ω is independent of $(\mathbf{s}, \mathbf{x}_\mathbb{C}, \mathbf{x}_\mathbb{P})$, the conditional distribution of $\mathbf{c} = \Omega \mathbf{x}_\mathbb{P}$ given $(\mathbf{s}, \mathbf{x}_\mathbb{C}, \mathbf{x}_\mathbb{P})$ is uniform on the sphere and does not depend on the condition, i.e., $\mathbf{c} \perp (\mathbf{s}, \mathbf{x}_\mathbb{C}, \mathbf{x}_\mathbb{P})$. By the chain rule

$$I(\mathbf{s}_\mathbb{P}; \mathbf{x}) = I(\mathbf{s}_\mathbb{P}; \mathbf{x}_\mathbb{C}, \mathbf{c}) = I(\mathbf{s}_\mathbb{P}; \mathbf{x}_\mathbb{C}) + I(\mathbf{s}_\mathbb{P}; \mathbf{c}|\mathbf{x}_\mathbb{C}) = I(\mathbf{s}_\mathbb{P}; \mathbf{x}_\mathbb{C}), \tag{15}$$

as $I(\mathbf{s}_\mathbb{P}; \mathbf{c}|\mathbf{x}_\mathbb{C}) = 0$ by $\mathbf{c} \perp (\mathbf{s}_\mathbb{P}, \mathbf{x}_\mathbb{C})$. Because $\mathbf{x}_\mathbb{C}$ is a function of $\mathbf{z}_\mathbb{C}$ for a given SNR, by the DPI: $I(\mathbf{s}_\mathbb{P}; \mathbf{x}_\mathbb{C}) \leq I(\mathbf{s}_\mathbb{P}; \mathbf{z}_\mathbb{C})$. An eavesdropper that observes $\mathbf{x}$ over a noisy channel cannot achieve more information than $I(\mathbf{s}_\mathbb{P}; \mathbf{x})$, also by the DPI. The same argument holds for encryption of $\mathbf{g} \in SO(d_g)$ with $\Omega \sim \mu_{SO(d_g)}$, where $\Omega\mathbf{g} \sim \mu_{SO(d_g)}$ for every fixed $\mathbf{g}$ by invariance of the Haar measure, so $I(\mathbf{c}; \mathbf{g}) = 0$.

Information leakage without constant power constraint. Let $\mathbf{x} \in \mathbb{R}^m$, $\mathrm{r} = ||\mathbf{x}||_2^2/m$ and $\mathbf{c} = \Omega\mathbf{x}$ with $\Omega \sim \mu_{O(m)}$ independent of $\mathbf{x}$. $||\Omega\mathbf{x}||_2 = ||\mathbf{x}||_2$, r is a function of $\mathbf{c}$, and by the argument above, given $\mathbf{x}$ the channel input $\mathbf{c}$ is uniform on the sphere of radius $\sqrt{m\mathrm{r}}$, i.e., it depends on $\mathbf{x}$ only through r. Then

$$I(\mathbf{x}; \mathbf{c}) = I(\mathbf{x}; \mathbf{c}, \mathrm{r}) = I(\mathbf{x}; \mathrm{r}) + I(\mathbf{x}; \mathbf{c}|\mathrm{r}) = I(\mathbf{x}; \mathrm{r}) = H(\mathrm{r}) - H(\mathrm{r}|\mathbf{x}) = H(\mathrm{r}), \tag{16}$$

where $H(\mathrm{r}|\mathbf{x}) = 0$ is due to r is a function of $\mathbf{x}$, and we assume that r is discrete. If r is not discrete, then $I(\mathbf{x}; \mathrm{r})$ is not bounded.

A.2 Implementation Details

The encoder and the decoder of both base models are the convolutional networks of Locatello et al. (2019) with $d = 10$, and the decoder of Lie-VAE takes the flattened $\mathbf{g}$. Ada-GVAE uses $\beta = 4$. Lie-VAE uses $\beta = 1$, $d_g = 10$. Base models are trained for 60 epochs and the channel coding for 40 epochs with AdamW, learning rate 10^{-3}, batch size 128 and gradient norm clipping at 100.

Each channel coding MLP has two hidden layers of 128 units with GELU activations and FiLM layers conditioned on the SNR in dB. The channel coding is trained with the objective of the base model, where the latent sample passes through the channel coding, encryption, the channel and decryption, and only the channel coding is updated.

The private part is selected with 4000 labelled samples. To enable selection, the mutual information between each latent component and each factor is estimated with a 20-bin histogram. For every private factor we select all latent components with at least $\tau = 0.5$ of the largest value, and the private part is the union over private factors.

The probes are trained on the decoded latent layer of 3000 labelled samples and tested on 1000 other samples. MIG and PSNR are computed for 3000 samples.

A.3 Additional Results

In this appendix we provide some additional plots. We plot mean and standard deviation over three seeds. Fig. 4 shows MIG and the reconstruction quality of the legitimate receiver vs SNR for Ada-GVAE and Lie-VAE base models. Fig. 5 shows secrecy vs SNR and Fig. 6 shows reconstructions with Lie-VAE base model.

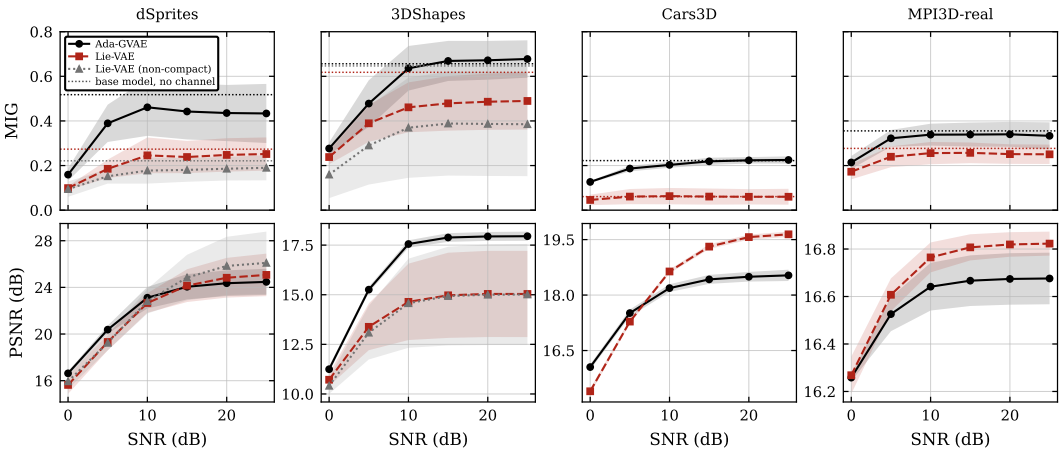


Figure 4: Reconstruction quality at the legitimate receiver vs SNR. Disentanglement quality vs SNR

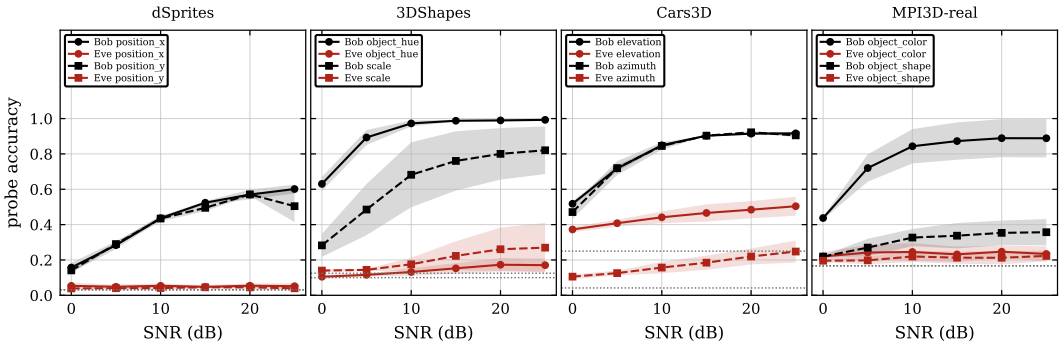


Figure 5: Accuracy of the best probe for each private factor vs SNR for the legitimate receiver (Bob, black) and the eavesdropper (Eve, red) with Lie-VAE base model. Dotted lines show the probability of a random guess.

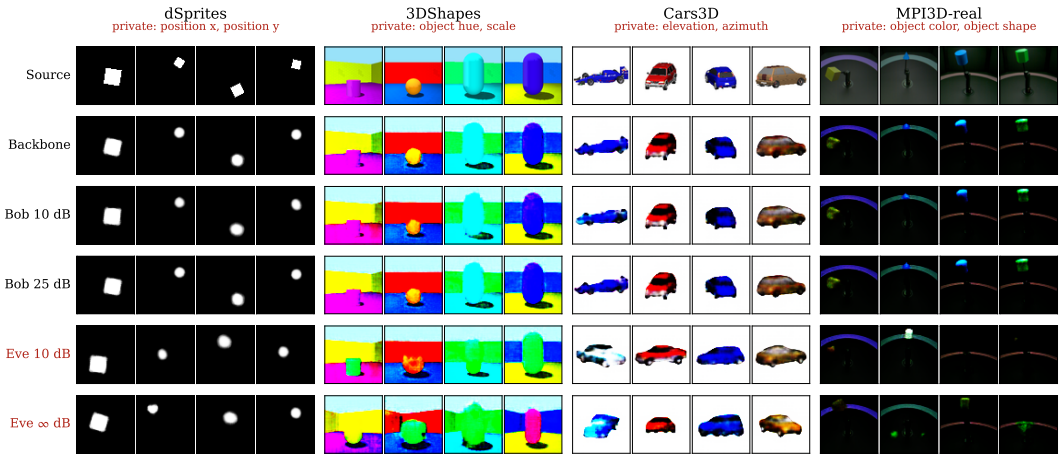


Figure 6: Reconstructions with Lie-VAE as a base model: base model without channel (Backbone), legitimate receiver (Bob) at 10 and 25 dB, and eavesdropper (Eve) at 10 dB and without channel noise (∞ dB).