Uncertainty-aware Unsupervised Video Hashing

Yucheng Wang
Texas A&M University

Mingyuan Zhou
University of Texas at Austin

Yu Sun
University of South Florida

Xiaoning Qian
Texas A&M University

Abstract

Learning to hash has become popular for video retrieval due to its fast speed and low storage consumption. Previous efforts formulate video hashing as training a binary auto-encoder, for which noncontinuous latent representations are optimized by the biased straight-through (ST) back-propagation heuristic. We propose to formulate video hashing as learning a discrete variational auto-encoder with the factorized Bernoulli latent distribution, termed as Bernoulli variational auto-encoder (BerVAE). The corresponding evidence lower bound (ELBO) in our BerVAE implementation leads to closed-form gradient expression, which can be applied to achieve principled training along with some other unbiased gradient estimators. BerVAE enables uncertainty-aware video hashing by predicting the probability distribution of video hash codewords, thus providing reliable uncertainty quantification. Experiments on both simulated and real-world large-scale video data demonstrate that our BerVAE trained with unbiased gradient estimators can achieve the state-of-the-art retrieval performance. Furthermore, we show that quantified uncertainty is highly correlated to video retrieval performance, which can be leveraged to further improve the retrieval accuracy. Our code is available at https://github.com/wangyucheng1234/BerVAE

1 INTRODUCTION

With the bursting of social media data, in particular from the video sharing services such as YouTube and TikTok, efficient search engines and recommendation systems for such high-volume data are crucial for diverse online services. Hashing is one of fast, stable, and accurate algorithms for content-based retrieval (Indyk and Motwani, 1998; Dasgupta et al., 2011; Broder, 1997; Broder et al., 1997). Traditionally, the design of the hash function that maps the input data to hashing keys or code-words (hash-codes) requires significant efforts to achieve the desired retrieval effectiveness and efficiency. Recent advances in learning to hash make it possible to learn this hash function automatically from data for complicated media data such as documents, images, and videos. As discussed in many previous works (Zhang et al., 2016; Song et al., 2018), for large-scale video databases, learning to hash for video retrieval can bring storage and computational benefits with the availability of high volume of streaming video data for training.

Modern deep neural networks (DNNs) have been applied to different computer vision tasks, including image classification (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015), semantic segmentation (Long et al., 2015; Ronneberger et al., 2015), image synthesis (Goodfellow et al., 2014; Kingma and Welling, 2013), and have achieved great successes. While those DNNs with modern computer hardware are capable of modeling high-dimensional and highly non-linear mappings and can even keep up with the human experts on a series of challenging tasks, they are notorious for making overconfident predictions. In some scenarios when designing DNNs for predictions with critical consequences, reliable uncertainty quantification is as important as accurate model prediction. The Bayesian methods, coming naturally with the capability of quantifying predictive uncertainty, have been integrated with many machine learning models to achieve better empirical performance and uncertainty quantification. Recent advances in different approximate Bayesian inference methods, for example, Monte-Carlo (MC) Dropout (Gal and Ghahramani, 2016) with corresponding amortized variational inference (Kingma and Welling, 2013), have further made it possible to scale the Bayesian learning methods to large DNN models.

Many previous works in video hashing train an auto-encoder with dichotomized latent representations as hash-codes for corresponding videos (Zhang et al., 2016; Song et al., 2018; Li et al., 2019b, 2021; Yuan et al., 2020). Al-

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

though they have achieved satisfying retrieval performance on large-scale datasets, there are several limitations to be addressed. First, the binary hash-code of each video is generated by dichotomizing the continuous latent representations, often in some heuristic ways. When training the auto-encoder, heuristic tricks (Bengio et al., 2013) were adopted to approximate the back-propagated gradients due to non-differentiability of the discrete binarization operators. This may affect the learning-to-hash performance. Second, none of them can provide reasonable uncertainty quantification. With potential uncertainty in derived video hashing due to the data size, data heterogeneity, as well as the data-driven nature of learning to hash, it is critical to have a new video hashing model that can reliably quantify the uncertainty, which is not available in any of the existing methods to the best of our knowledge.

In this work, we directly model hash-codes probabilistically as factorized latent Bernoulli random vectors. Learning to hash can be formulated by deriving the variational posterior of this latent random vector in a Bernoulli variational auto-encoder, hence the name BerVAE. With the learned variational posterior of hash-codes, BerVAE is capable to provide high-quality uncertainty quantification. To train our BerVAE considering the discrete latent Bernoulli random vector, we adopt and benchmark several gradient estimators, including the commonly adopted straight-through (ST) heuristic, which ignores the discrete binarization operator, as well as several biased and unbiased gradient estimators when involving discrete random variables: Gumbel-Softmax (GS) (Jang et al., 2016; Maddison et al., 2016), and unbiased uniform gradient (U2G) (Yin et al., 2020). U2G, independently developed as DisARM (Dong et al., 2020), is an improved version of the ARM gradient estimator (Yin and Zhou, 2019). We also derive the closed-form gradient with our adopted decoder in video hashing. We perform comprehensive ablation studies on both synthetic and real-world data and demonstrate that our BerVAE-based video hashing can achieve state-of-the-art retrieval performance; and more importantly, BerVAE facilitates uncertainty quantification of derived hash-codes for better video retrieval and consequent decision making.

2 RELATED WORKS

We review the related works on *learning to hash* and *generative modeling with discrete latent representations*.

Pioneering works of learning to hash include Spectral Hashing (Weiss et al., 2008), Linear Discriminant Analysis (LDA) Hashing (Strecha et al., 2011), and Graph Hashing (Liu et al., 2011, 2014). Although those methods can automate the design of hash functions using the collected data and save human efforts, they often require handcraft features. Deep learning-based hashing models have been proposed recently and had success in analyzing various types of data. Semantic Hashing (Salakhutdinov and Hinton, 2009) is among the earliest works using DNNs for hashing, where a two-stage procedure is proposed to train a deep auto-encoder for document retrieval in a fully unsupervised manner.

For high-dimensional complex data such as images and videos, some previous deep hashing models include Deep Hashing (DH) (Erin Liong et al., 2015), Deep Pairwise-Supervised Hashing (DPSH) (Li et al., 2016), and Self-Supervised Temporal Hashing (SSTH) (Zhang et al., 2016). DH (Erin Liong et al., 2015) is capable to capture the non-linearity of the learned hashing function and the predicted hash-codes of the whole database can maintain the desired variability and balance by the designed activation function and training losses. The authors of DPSH (Li et al., 2016) further introduced the pairwise contrastive labels to the deep-hashing model training. To model the temporal order information of video frame sequences, the authors in SSTH (Zhang et al., 2016) proposed a Binary Long Short-Term Memory (LSTM) encoder and a Recurrent Neural Network (RNN) decoder for video hashing. More recent advances, including Self-Supervised Video Hashing (SSVH) (Song et al., 2018), Neighborhood Preserving Hashing (NPH) (Li et al., 2019b), Central Similarity Quantization (CSQ) (Yuan et al., 2020), Unsupervised Variational Video Hashing (UVVH) (Li et al., 2019a) and Bidirectional Transformers Hashing (BTH) (Li et al., 2021), further improved the video retrieval accuracy through novel neural network architectures (Song et al., 2018; Li et al., 2019b, 2021), loss function design (Song et al., 2018; Li et al., 2019b) and unsupervised contrastive label generation (Li et al., 2019a; Yuan et al., 2020; Li et al., 2019b).

VAE (Kingma and Welling, 2013), a deep generative model suitable for unsupervised representation learning, models the latent representations as random variables. When training VAEs, the evidence lower bound (ELBO) is optimized to minimize the discrepancy between the generative distribution and data distribution. Discrete generalization models of VAE have been proposed recently and are successful for data compression (van den Oord et al., 2017; Razavi et al., 2019) and discrete representation learning (Rolfe, 2016; Park et al., 2021). However, optimizing these discrete VAEs is notoriously challenging as the reparameterization tricks can not be directly applied and the gradient of the latent distribution is hard to estimate. Some popular biased gradient estimators includes commonly adopted straight-through (ST) (Bengio et al., 2013), which ignores the discontinuity by the introduction of discrete latent distribution, and Gumbel-Softmax (GS) (Jang et al., 2016; Maddison et al., 2016), which relaxes the discrete latent representations continuously with the softmax function. REINFORCE, a widely applicable unbiased gradient estimator that estimates gradients by Monte Carlo estimation, often suffers from high variance, and many recent works

were proposed to alleviate this issue (Yin and Zhou, 2019; Dong et al., 2020; Yin et al., 2020). In particular, VAEs with binary latent representations are especially suitable for unsupervised learning to hash, which was applied to text retrieval (Chaidaroon and Fang, 2017; Dong et al., 2019; Dadaneh et al., 2020; Mena and Ñanculef, 2019) as well as image and video retrieval (Verwilst et al., 2021; Fajtl et al., 2020; Li et al., 2019a). One drawback is that previous works reparameterize the latent Bernoulli distribution using continuous distribution or thresholding function, whose quantization error may result in information loss. Moreover, none of them provides reasonable uncertainty quantification, which is crucial for reliable data retrieval.

The major difference between our Bernoulli VAE (BerVAE) and previous models that incorporate VAEs with binary latent representations is that we optimize the distribution of discrete hash-codes using the U2G/DisARM estimator, which can alleviate the bias issue and have low variance; and moreover, our model can provide reasonable and reliable uncertainty quantification for video retrieval, which has not been discussed in any previous works to the best of our knowledge.

3 METHOD

3.1 Problem Settings

Suppose that we have a video dataset with N video clips $\boldsymbol{S} = \{V_1, V_2, V_3, \dots, V_N\}$. The objective of unsupervised video hashing is to map each input video to the corresponding k-bit hash-codes $\mathcal{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_N | \boldsymbol{b}_i \in \{-1, 1\}^k\}$ such that each video pair V_i, V_j with similar content can have the hash-codes $\boldsymbol{b}_i, \boldsymbol{b}_j$ with a smaller Hamming distance while a video pair V_p, V_q with different contents will have $\boldsymbol{b}_p, \boldsymbol{b}_q$ with a larger Hamming distance between them.

To enable uncertainty-aware video retrieval, we here innovate a Bayesian video hashing strategy by formulating unsupervised video hashing to find the posterior distribution of $\mathcal{B}$ given $\boldsymbol{S}$: $p(\mathcal{B}|\boldsymbol{S}) = \frac{p(\boldsymbol{S}|\mathcal{B})p(\mathcal{B})}{p(\boldsymbol{S})}$. Since $p(\boldsymbol{S}) = \int_{\mathcal{B}} p(\boldsymbol{S}|\mathcal{B})p(\mathcal{B})d\mathcal{B}$ is intractable, we reformulate the problem as minimizing the Kullback–Leibler (KL) divergence between the variational distribution $q_\phi(\mathcal{B}|\boldsymbol{S})$, which is parameterized by ϕ, and the true posterior distribution $p(\mathcal{B}|\boldsymbol{S})$ to approximate. This variational inference problem can be solved by minimizing the negative evidence lower bound (negative ELBO) given the training video data (Kingma and Welling, 2013). If the likelihood $p(\boldsymbol{S}|\mathcal{B})$ is parameterized by ψ, then the negative ELBO given $\boldsymbol{S}$ can be written as follows:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\phi, \psi) = &-\mathbb{E}_{q_\phi(\mathcal{B}|\mathcal{S})}[\log p_\psi(\boldsymbol{S}|\mathcal{B})] \\ &+ D_{KL}(q_\phi(\mathcal{B}|\boldsymbol{S})||p(\mathcal{B})).\end{aligned} \tag{1}$$

3.2 BerVAE for Video Hashing

Figure 1 illustrates our BerVAE-based video hashing architecture. To represent a video clip, we randomly choose M frames from each video and the temporal information of frame-sequence is exploited by a transformer network, following Li et al. (2021). Denote the output of the transformer encoder network for video V_i by $\{\boldsymbol{h}_i^1, \boldsymbol{h}_i^2, \dots, \boldsymbol{h}_i^M | \boldsymbol{h}_i^m \in \mathbb{R}^{d\times 1}\}$. We use a linear layer with a weight matrix W^t and a mean pooling layer to aggregate them into a vector with the same dimension of hash-code $\boldsymbol{b}_i$, parametrizing the corresponding success probabilities of random hash-code bits:

$$\begin{aligned}&\boldsymbol{t}_i^m = W^t \boldsymbol{h}_i^m \in \mathbb{R}^{k\times 1}, m = 1, 2, \dots, M, \\ &\bar{\boldsymbol{t}}_i = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{t}_i^m, \\ &q_\phi(\boldsymbol{b}_i^v | V_i) = \begin{cases}\sigma(\bar{\boldsymbol{t}}_i^v) \text{ when } \boldsymbol{b}_i^v = 1 \\ 1 - \sigma(\bar{\boldsymbol{t}}_i^v) \text{ when } \boldsymbol{b}_i^v = -1\end{cases}, v = 1, 2, \dots, k,\end{aligned} \tag{2}$$

where $\boldsymbol{b}_i^v$ denotes the v-th bit of $\boldsymbol{b}_i$, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. As we are reconstructing the video feature $V_i \in \mathbb{R}^{M\times d}$ from the aggregated binary vector $\boldsymbol{b}_i \in \mathbb{R}^{k\times 1}$, we use $\boldsymbol{w}_{\text{Rc1}} \in \mathbb{R}^{M\times 1}$ to reconstruct the whole frame-sequence, and $W_{\text{Rc2}} \in \mathbb{R}^{k\times d}$ to reconstruct the features of each frame $V_i = \{\boldsymbol{v}_i^1, \boldsymbol{v}_i^2, \dots \boldsymbol{v}_i^M\}$. Let $\boldsymbol{w}_{\text{Rc1}}^m$ denote the m-th entry of $\boldsymbol{w}_{\text{Rc1}}$. We assume that the likelihood of video features $\boldsymbol{v}_i^m$ given hash-code $\boldsymbol{b}_i$ is a Gaussian distribution with mean $\boldsymbol{\mu}_i^m = \boldsymbol{w}_{\text{Rc1}}^m \boldsymbol{b}_i^T W_{\text{Rc2}}$ and isotropic diagonal covariance matrix $\boldsymbol{\Sigma} = \sigma_{\text{Rc}}^2 \boldsymbol{I}_{d\times d}$. The log-likelihood of $\boldsymbol{S}$ with respect to $\mathcal{B}$ is given by:

$$\log p_\psi(\boldsymbol{S}|\mathcal{B}) = \sum_{i=1}^{N}\sum_{m=1}^{M} \log p_\psi(\boldsymbol{v}_i^m | \boldsymbol{b}_i) = L_v + C, \tag{3}$$

where the sum of squared errors $L_v = -\sum_{i=1}^{N}\sum_{m=1}^{M} \frac{1}{2\sigma_{\text{Rc}}^2}||\boldsymbol{v}_i^m - \boldsymbol{\mu}_i^m||_2^2$ and $C = -\sum_{i=1}^{N} \frac{dM}{2}\log 2\pi\sigma_{\text{Rc}}^2$ is a constant. The negative ELBO in (1) can be rewritten as follows:

$$\begin{aligned}\mathcal{L}_{ELBO}(\phi, \psi) = &\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\sum_{i=1}^{N}\sum_{m=1}^{M} \frac{1}{2\sigma_{\text{Rc}}^2}||\boldsymbol{v}_i^m - \boldsymbol{\mu}_i^m||_2^2] \\ &+ \sum_{i=1}^{N} D_{KL}(q_\phi(\boldsymbol{b}_i|V_i)||p(\boldsymbol{b}_i)).\end{aligned} \tag{4}$$

Modeling the prior of each hash-code bit $p(\boldsymbol{b}_i^v)$ as a Bernoulli distribution with the success probability $\boldsymbol{p}^v$ to be 1 and otherwise $1 - \boldsymbol{p}^v$ to be -1, the KL term has a closed form $D_{KL}(q_\phi(\boldsymbol{b}_i|V_i)||p(\boldsymbol{b}_i)) = \sum_{v=1}^{k}[\sigma(\bar{\boldsymbol{t}}_i^v)\log\frac{\sigma(\bar{\boldsymbol{t}}_i^v)}{\boldsymbol{p}^v} + (1 - \sigma(\bar{\boldsymbol{t}}_i^v))\log\frac{1-\sigma(\bar{\boldsymbol{t}}_i^v)}{1-\boldsymbol{p}^v}]$.

In practice, in order to balance the scales of two loss terms, we adopt the following minimization objective:

$$\mathcal{L}_{ELBO}(\phi,\psi) = \frac{\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\sum_{i=1}^{N}\sum_{m=1}^{M}||\boldsymbol{v}_i^m - \boldsymbol{\mu}_i^m||_2^2]}{MNd} + \frac{\lambda\sum_{i=1}^{N} D_{KL}(q_\phi(\boldsymbol{b}_i|V_i)||p(\boldsymbol{b}_i))}{KN}, \tag{5}$$

where λ is a hyper-parameter reflecting our prior belief relative to the likelihood of observed data.

3.3 Training BerVAE

To train BerVAE, we need to back-propagate the gradients of the expected negative ELBO with respect to the encoder and decoder parameters $\{\phi, \psi\}$. With the discrete random hash-code $\boldsymbol{b}_i$, the encoder parameters ϕ can not be optimized directly using the back-propagation algorithm with Monte-Carlo sampling. In previous works, the encoder network is trained using the straight-through (ST) heuristic, ignoring the discrete thresholding when applying the chain rule.

Principled training to achieve tighter ELBO is crucial for both Bayesian inference and uncertainty quantification to derive effective video retrieval by better capturing the generative distribution of hash-codes. We here consider gradient estimators that are unbiased and/or with low variance and couple them in back-propagation for training BerVAE. In particular, we apply unbiased uniform gradient (U2G) (Yin et al., 2020), which achieves the minimum variance of augment-reinforce-merge (ARM) estimators (Yin and Zhou, 2019) by integrating out the induced randomness by reparameterizing discrete random variables.

For any function $f : \{-1, 1\}^k \to \mathbb{R}$, and Bernoulli random vector $\boldsymbol{b} \sim q_\phi(\boldsymbol{b}|S)$ as in (2), the U2G estimator for the gradient of the form $\nabla_{\bar{\boldsymbol{t}}}\mathbb{E}_{\boldsymbol{b}\sim q_\phi(\boldsymbol{b})}[f(\boldsymbol{b})]$ is:

$$\nabla_{\bar{\boldsymbol{t}}}\mathbb{E}_{\boldsymbol{b}\sim q_\phi(\boldsymbol{b})}[f(\boldsymbol{b})] \approx \frac{\sigma(|\bar{\boldsymbol{t}}|)}{2}[(f(2\times\mathbf{1}_{[\boldsymbol{u}>\sigma(-\bar{\boldsymbol{t}})]}-\mathbf{1}) - f(2\times\mathbf{1}_{[\boldsymbol{u}<\sigma(\bar{\boldsymbol{t}})]}-\mathbf{1}))(\mathbf{1}_{[\boldsymbol{u}>\sigma(-\bar{\boldsymbol{t}})]}-\mathbf{1}_{[\boldsymbol{u}<\sigma(\bar{\boldsymbol{t}})]})], \tag{6}$$

where $\boldsymbol{u}$ is sampled from a factorized continuous uniform distribution between 0 and 1, and $\mathbf{1}_{[\boldsymbol{u}>\sigma(-\bar{\boldsymbol{t}})]} = (\mathbf{1}_{[\boldsymbol{u}^1>\sigma(-\bar{\boldsymbol{t}}^1)]}, \mathbf{1}_{[\boldsymbol{u}^2>\sigma(-\bar{\boldsymbol{t}}^2)]}, \ldots, \mathbf{1}_{[\boldsymbol{u}^k>\sigma(-\bar{\boldsymbol{t}}^k)]})^T$. One U2G sample requires two evaluations of $f(\cdot)$, compared to 2^k evaluations if we want to calculate the closed-form gradient $\nabla_{\bar{\boldsymbol{t}}}\mathbb{E}_{\boldsymbol{b}\sim q_\phi(\boldsymbol{b})}[f(\boldsymbol{b})]$ for arbitrary $f(\cdot)$. To train our BerVAE, the gradient with respect to the parameters ϕ of the q distribution, $\nabla_\phi\mathbb{E}_{\boldsymbol{b}\sim q_\phi(\boldsymbol{b})}[f(\boldsymbol{b})]$, can be further coupled with back-propagation.

As a special case with a linear decoder network and factorized Gaussian likelihood, we can calculate the closed-form negative ELBO and the gradient of negative ELBO with respect to the encoder parameters by back-propagation without evaluating the decoder for 2^k times. We include the full derivation in *Appendix* A. The gradient contributed from the KL term and $\frac{\partial\bar{\boldsymbol{t}}}{\partial\boldsymbol{\phi}}$ can be calculated with the standard back-propagation procedure. We also benchmark the U2G estimator and our derived closed-form gradients with other commonly adopted gradient estimators, including ST and Gumbel-Softmax (GS) estimators (Jang et al., 2016; Maddison et al., 2016) in Section 4.1.

3.4 Uncertainty-aware Video Retrieval

The probabilistic modeling of hash-codes enables learning to hash with Bernoulli latent vectors as well as quantifying their uncertainty. We predict the hash-code of any given video clip by thresholding each entry of the predicted success probability:

$$\boldsymbol{b}_i^v = \begin{cases} 1 & \text{if } \sigma(\bar{\boldsymbol{t}}_i^v) \geq 0.5, \\ -1 & \text{otherwise.} \end{cases} \tag{7}$$

This is equivalent to the *maximize-a-posterior* (MAP) inference with the variational posterior distribution $q_\phi(\boldsymbol{b}_i|V_i)$. For each query video, we retrieve the most similar videos by the Hamming distances of their hash-codes. We adopt Shannon's entropy of the variational posterior distribution $q_\phi(\boldsymbol{b}_i|V_i)$ to quantify the inferred hash-code uncertainty:

$$H(q_\phi(\boldsymbol{b}_i|V_i)) = -\sum_{v=1}^{k}[\sigma(\bar{\boldsymbol{t}}_i^v)\log\sigma(\bar{\boldsymbol{t}}_i^v) + (1-\sigma(\bar{\boldsymbol{t}}_i^v))\log(1-\sigma(\bar{\boldsymbol{t}}_i^v))]. \tag{8}$$

This quantified uncertainty can be indicative of the video retrieval error and help further improve retrieval performances. As a simple illustration in this work, we consider withholding the videos with the highest quantified uncertainty of latent hash-codes from the final video retrieval results. Our experiments in Section 4 demonstrate empirically that this straightforward strategy can effectively improve video retrieval accuracy and serve as a way to benchmark the uncertainty quantification performance in video hashing.

4 EXPERIMENTS

We first evaluate different gradient estimators for model training and demonstrate the effectiveness of BerVAE-based hash-code embedding with low-dimensional toy examples in Section 4.1. We then evaluate BerVAE-based video hashing on real-world large-scale video datasets. We introduce the FCVID dataset, the evaluation metrics, and training details in Sections 4.2, 4.3 and 4.4. We compare

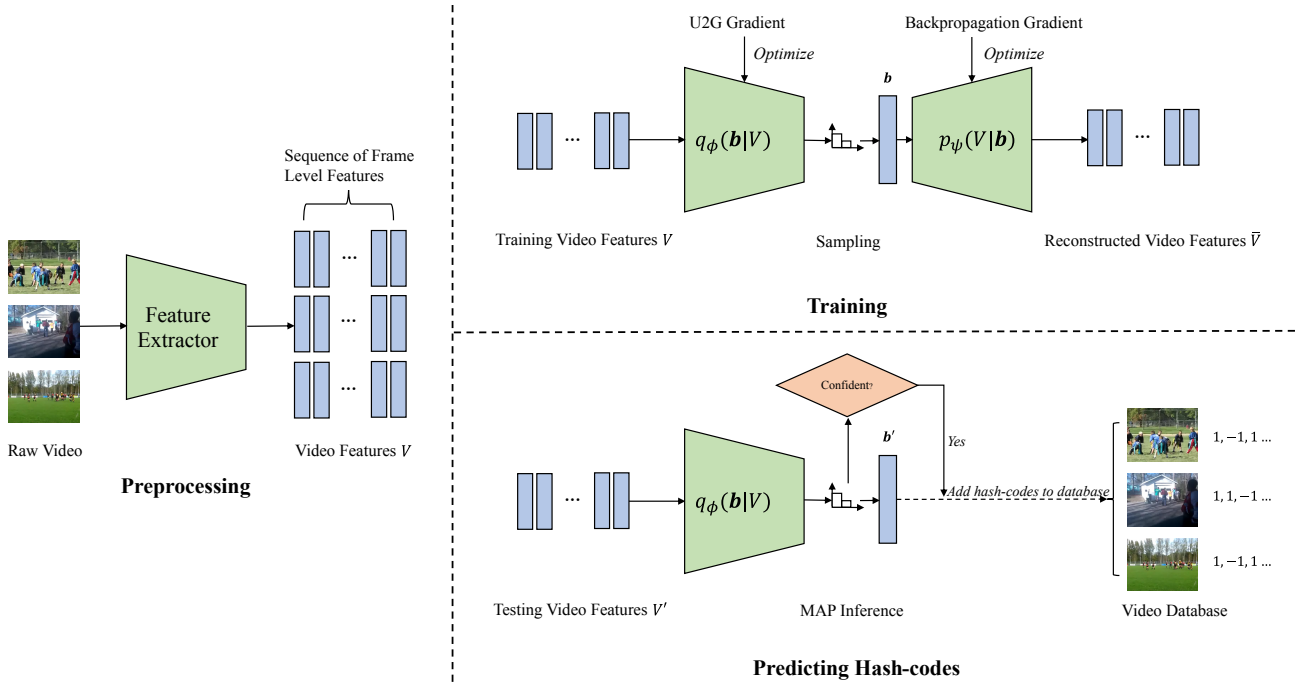


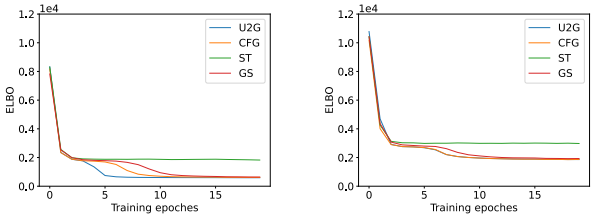
Figure 1: Schematic illustration of BerVAE-based video hashing: hash-codes are modeled as latent Bernoulli random vectors for uncertainty-aware unsupervised video hashing.

our method with the baseline model (Li et al., 2021), from which we inherit the backbone architecture with, and show the uncertainty quantification capability of BerVAE in Sections 4.5 and 4.6 on FCVID. We have also conducted experiments on the ActivityNet dataset (Heilbron et al., 2015), reported in *Appendix* C.2 due to limited space. Lastly in Section 4.7, we visualize our uncertainty quantification results qualitatively and through t-SNE visualization.

4.1 Toy Examples

Experimental Settings Both the training and test sets in our first toy example are composed of 10,000 points independently sampled from a mixture of 2-D Gaussian distributions with the mean $\boldsymbol{\mu}_1 = (-0.5, 0)^T$, $\boldsymbol{\mu}_2 = (0.5, 0)^T$, and $\boldsymbol{\mu}_3 = (1.5, 0)^T$, and variance $\sigma^2 I$ of different σ values. The encoder network is composed of three fully connected layers, each with the ReLU activation function and batch normalization layer. The decoder network is composed of one linear layer. We follow the same model training as described in Section 3 with the 2-D sampled Gaussian inputs. The latent vector z is modeled as a 2-bit Bernoulli random vector whose success probabilities are modeled by the encoder network with a sigmoid activation function. Similar as Dong et al. (2020), we compare different gradient estimators by their achieved training ELBO.

Results & Discussion We report the training ELBO, derived hash-code latent representations, and predicted uncertainty of each test points in Figures 2, 3, and 4. We use the same network initialization and training batches for all the experiments.

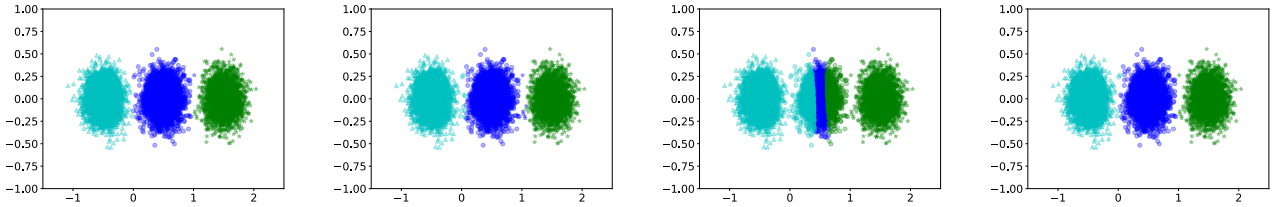


(a) $\sigma = 0.15$ (b) $\sigma = 0.3$

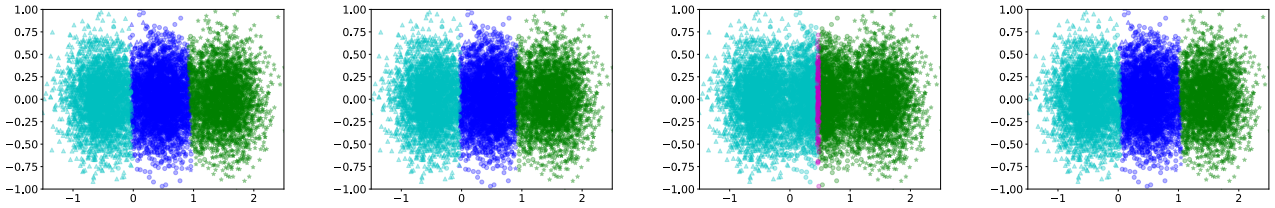
Figure 2: Training ELBO for BerVAE with respect to the training epochs using different gradient estimators.

In Figure 2, we report our training ELBO with respect to the training epochs, where the training ELBO is evaluated using the closed-form ELBO derived in *Appendix* A. We observe that model training based on the ST gradient estimator typically has the worst training loss among all of the tested gradient estimators, probably because it has the largest estimation bias. The convergence rate of training using U2G is slower than the one with the Closed-Form Gradient (CFG), and the training by the Gumbel-Softmax (GS) estimator is even slower. While training based on U2G or GS converges slower than using CFG, they can be easily implemented and applied to other discrete VAEs with any decoder network architectures.

To benchmark and demonstrate the effectiveness of de-

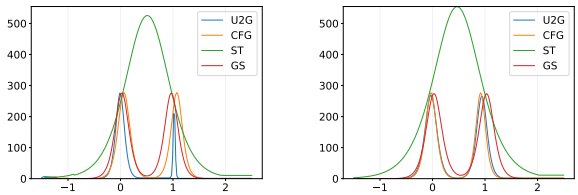


(a) U2G, $\sigma = 0.15$ (b) CFG, $\sigma = 0.15$ (c) ST, $\sigma = 0.15$ (d) GS, $\sigma = 0.15$



(e) U2G, $\sigma = 0.3$ (f) CFG, $\sigma = 0.3$ (g) ST, $\sigma = 0.3$ (h) GS, $\sigma = 0.3$

Figure 3: The clustering results based on hash-codes by BerVAE trained with different gradient estimators.



(a) $\sigma = 0.15$ (b) $\sigma = 0.3$

Figure 4: The predicted uncertainty of the BerVAE trained with different gradient estimators.

rived hash-codes by our BerVAE, Figure 3 visualizes the clustering results of hash-codes from BerVAE trained with different gradient estimators. The points generated from different Gaussian mean values are labeled with the markers of different shape. The points encoded into $\{-1,-1\}, \{-1,1\}, \{1,-1\}$ and $\{1,1\}$ are colored in blue, green, cyan, and magenta, respectively. The clustering results by BerVAE trained using ST are much worse than the ones with other gradient estimators. In all of the experiments with either CFG, U2G, or GS, most of the points encoded into the same hash-code are generated from the same Gaussian mixture component. The points in cyan and green are far from each other compared to the points in blue and green. Clearly, the distance relationships in the original input space are well preserved in terms of the Hamming distance of their corresponding hash-codes. This indicates that effective retrieval can be achieved by comparing the Hamming distance of the derived hash-codes by BerVAE.

As discussed in Section 3.4, we can quantify the uncertainty of the derived latent hash-codes by the Shannon's entropy of the predicted posterior distribution in BerVAE. With the same y-coordinate of each Gaussian mixture component center in this example, we integrate the Shannon's entropy along the y-axis and plot the uncertainty with respect to the x-coordinate in Figure 4 to illustrate the uncertainty quantification capability of BerVAE. From the plots in Figure 4, the predicted uncertainty is the highest at the boundary between each of the neighboring clusters. The predicted uncertainty peaks of the CFG, U2G and GS estimators are closer to 0 and 1, and the x-coordinate values of the corresponding Gaussian mixture component centers. It is clear that the clustering results by ST are the worst and the predicted uncertainty is the most biased. Especially when $x \in [0, 1]$, the derived latent hash-codes from BerVAE with ST fail to cluster reasonably. Besides the above experiment, We provide another toy example, for which we set different y-coordinates for the centers of the Gaussian mixture components. We include our results in *Appendix* B due to limited space, which show the same trends as the first example. Based on these, we choose to optimize the encoder network of BerVAE by U2G for video hashing.

4.2 Real-world Video Dataset

FCVID, the Fudan-Columbia Video Dataset (Jiang et al., 2018), is a large-scale video dataset containing a total of 91,223 videos in 239 categories with a total duration of 4,232 hours. Following previous works (Zhang et al., 2016; Song et al., 2018; Li et al., 2019b,a, 2021; Yuan et al., 2020), we use 91,185 videos of them with 45,585 for training and 45,600 for testing.

4.3 Evaluation Metrics

Video Retrieval In a ranking-based retrieval system, Average Precision (AP) is the integrated performance measure jointly considering precision and recall. Let $P(k)$ be the

precision of the top k videos, and rel(k) the indicator function that the k-th video is relevant, and $|\{\text{Relevant}\}|$ is the number of all the relevant videos in the database. We have:

$$AP(k) = \frac{\sum_{k=1}^{N} P(k) \times \text{rel}(k)}{|\{\text{Relevant}\}|}. \quad (9)$$

Following previous works (Zhang et al., 2016; Song et al., 2018; Li et al., 2019b,a, 2021; Yuan et al., 2020), we use the *mean Average Precision@*K (mAP@K) on the whole test set to evaluate the retrieval performance of BerVAE-based video hashing, with K retrieved video clips when we calculate the average precision.

Uncertainty Quantification In Section 3.4, we have discussed a strategy to withhold the uncertain videos from retrieval results to improve the accuracy of video retrieval. To compare the uncertainty quantification performance of different models in video retrieval, we investigate the *Improvement by Deleting Uncertain data* (IDU), based on the following expression:

$$\text{IDU}(K) = \int_0^1 [\text{mAP@}K(p) - \text{mAP@}K(0)]\text{d}p, \quad (10)$$

where mAP@$K(p)$ represents mAP@K after deleting p percentage of videos with the most uncertain inferred hash-codes. IDU measures how much the retrieval accuracy can be improved with the videos of the most uncertain hash-codes gradually removed from the database.

4.4 Backbone Architecture & Settings

Following Li et al. (2021), we use the Transformer network (Vaswani et al., 2017) to model the temporal order information of frame-sequences. For each video, we randomly select $M = 25$ frames and the dimension of the feature representation of each frame is 4,096. We use Adam (Kingma and Ba, 2014) and set the learning rate to 3×10^{-4}. We train our model for 200 epochs. For KL-loss, we set the prior strength of our BerVAE λ to be 0.1 and add $\epsilon = 1 \times 10^{-8}$ on both denominator and numerator to prevent numerical issues. We run the experiments on PyTorch (Paszke et al., 2019) 1.8.1 and use the same model parameter initialization for BTH (Li et al., 2021) and our BerVAE in each experiment. We do not control the randomness of SSVH (Song et al., 2018) as its implementation is based on Theano (The Theano Development Team et al., 2016).

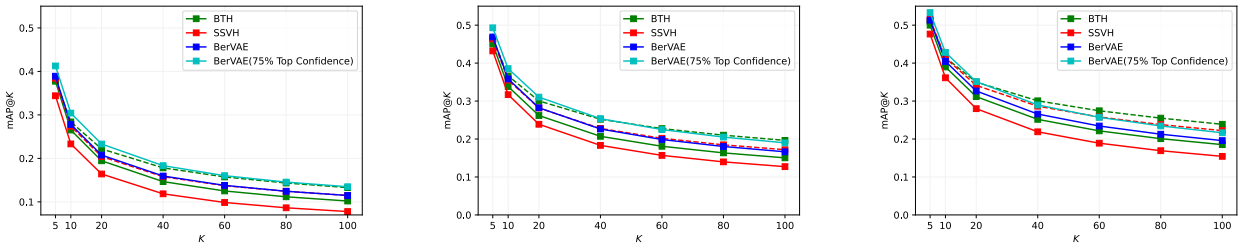
4.5 Comparison with Baselines

We compare the retrieval performance of our BerVAE-based video hashing with several state-of-the-art (SOTA) models whose implementations are open-source. Those SOTA models we choose are all trained with combinations of different loss terms, including commonly adopted pairwise contrastive loss. As our focus is unsupervised video hashing and demonstrating the advantages of principled training with BerVAE. For fair comparison, we choose to compare our BerVAE with the models trained with only the reconstruction loss. We have further tested other settings and included the results of our BerVAE trained with contrastive and other regularization loss terms in *Appendix* C.1, which also demonstrate superior performance compared to their corresponding baselines. The retrieval accuracy of the video clips with the top 75% confident hash-codes predicted by BerVAE is reported along with results of other models. We also include the results of our reproduced baseline model, SSVH (Song et al., 2018) and BTH (Li et al., 2021), with their original implementations including contrastive loss for completeness. Our experimental results on FCVID are reported in Figure 5, where the performances of the baselines trained with combined losses are plotted in dotted lines.

Observing the performance trends of the solid lines focusing on unsupervised video hashing in Figure 5, our BerVAE outperforms the baseline models on FCVID, and also on ActivityNet when we learn 32- and 64-bit hash-codes (for the latter, we leave the details in *Appendix* C.2). The baseline BTH (Li et al., 2021) trained with all the original loss terms still maintains performance advantage over BerVAE. However, when comparing with the other SOTA models with unsupervised hashing implementations, it is clear that our BerVAE-based video hashing achieves better or similar retrieval performances. Compared to the baseline trained with only the reconstruction loss, BerVAE significantly outperforms SSVH and achieves 0.014, 0.02 and 0.015 better mAP@20 with 32-, 64- and 128-bit hash-codes than BTH. BerVAE even achieves better performance than SSVH with the combined losses with 32- and 64-bit hash-codes. With the 25% uncertain videos withheld from retrieval, BerVAE can achieve similar performance with the BTH trained with combined losses. The full quantitative experimental results can be found in *Supplementary Tables* 1, 2, and 3 in *Appendix* C.1. We leave the design of uncertainty-preserving contrastive loss and other heuristic loss terms implemented in BTH for our future research for supervised hashing. One critical advantage of BerVAE is its uncertainty quantification capability. By withholding the videos with highly uncertain hash-codes, our uncertainty-aware video retrieval can achieve similar, if not better, retrieval performances with the models equipped with different loss terms.

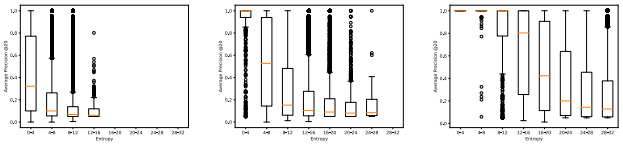
4.6 Uncertainty-aware Video Retrieval

To demonstrate the uncertainty quantification capability of BerVAE, we first include the box-plot of each query video's AP@20 with respect to the predicted uncertainty of its hash-code on FCVID in Figure 6. It is clear that, as the



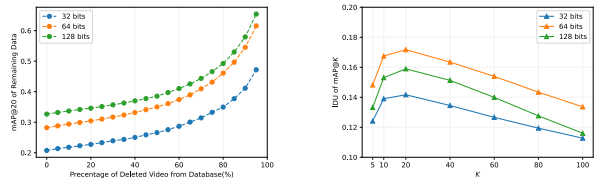
(a) 32-bit (b) 64-bit (c) 128-bit

Figure 5: Retrieval accuracy of BerVAE compared with state-of-the-art models with different bit length of hash-codes on FCVID. K is the number of retrieved video clips when calculating mAP@K given a query video.



(a) 32-bit (b) 64-bit (c) 128-bit

Figure 6: Box-plots of AP@20 values with respect to the predicted uncertainty of video hash-codes on FCVID.



(a) (b)

Figure 7: (a) mAP@20 of remaining video clips with respect to the percentage of clips withheld from retrieval; (b) IDU of mAP@K with different K.

predicted uncertainty of the video hash-code goes higher, its retrieval accuracy tends to get worse, illustrating a significant correlation between the predicted hash-code uncertainty and video retrieval error. Such trends can also be observed on ActivityNet, as provided in *Appendix* C.2.

Taking advantage of this observation, we further report mAP@20 for retrieved video clips when we withhold specific percentages of videos with uncertain hash-codes in Figure 7a. When video clips with the most uncertain hash-codes are withheld from retrieval results, we observe significantly improved average retrieval accuracy. In Figure 7b, we report the IDU values defined in (10), which are the integrated mAP improvement in Figure 7a, to illustrate the benefits of BerVAE-enabled uncertainty quantification. For all different K and hash-code bit-length settings, our model can consistently improve the retrieval accuracy. The quantitative details are provided in *Supplementary Table* 7 in *Appendix* C.1.

4.7 Visulization and Discussion

We visualize the retrieval and uncertainty quantification capability of a 64-bit BerVAE trained on FCVID. Figure 8 shows several exemplar frames of the video clips with the most certain and uncertain hash-codes. The uncertainty of inferred hash-codes differs drastically in different categories. For example, the clips in the "billiard" category have significantly lower uncertainty than clips in other categories. A possible explanation is that most of the "billiard" videos contain common visual features, for example a pool table, which makes them easier to identify than clips from other categories.

We have also investigated retrieval results using the query videos with the most certain and uncertain hash-codes in three different categories, whose exemplar frames are provided in Figure 9. Even within the same category, the uncertainty may differ by each video. Typically, the video clips with the most certain hash-codes have similar appearance as many other videos in the same category while the most uncertain videos look drastically different from other videos. Query videos with certain hash-codes often have more accurate retrieval results in general.

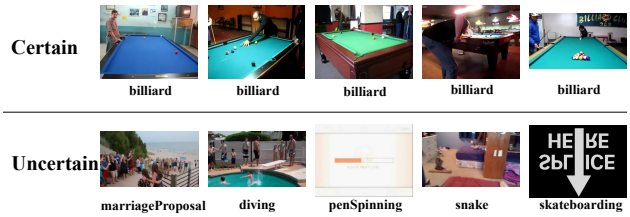
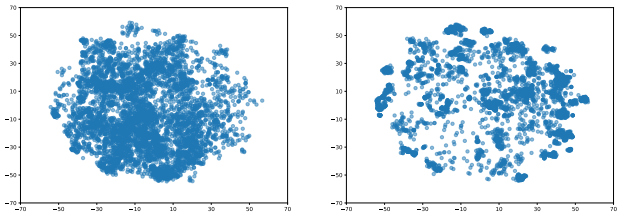


Figure 8: Exemplar frames of the five video clips with the most certain and uncertain hash-codes in FCVID.

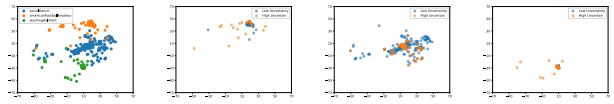
We also provide t-SNE visualization (Van der Maaten and Hinton, 2008) of the predicted hash-codes of video clips with different uncertainty levels and in different categories in Figures 10 and 11. It is obvious that for videos in the whole dataset and for videos in each category, compared to the certain hash-codes, the uncertain hash-codes are more spread-out in the latent hash-code space. They also tend to have larger distances from the hash-codes of the videos

Figure 9: The top-3 retrieved video clips of the query videos with the most certain and uncertain hash-codes in three selected categories of FCVID with exemplar frames.



(a) (b)

Figure 10: t-SNE visualization of hash-codes with the low (a) and high (b) uncertainty in FCVID: (a) and (b) visualize the hash-codes with the uncertainty ranked the top and bottom eighth, respectively.



(a) (b) (c) (d)

Figure 11: t-SNE visualization of video hash-codes of three categories in Figure 9: (a) video hash-codes from all three categories. (b-d) video hash-codes with the uncertainty ranked the top and bottom fifth (colored in blue and orange) in each of three categories, respectively.

in the same categories. Compared to clustered hash-codes with lower uncertainty, it is more likely for them to retrieve videos with different semantic meaning, resulting in high retrieval error rate.

5 SUMMARY AND FUTURE WORKS

We have developed a new Bayesian video hashing framework based on BerVAE, which captures the randomness of hash-codes by factorized latent Bernoulli distributions. Our BerVAE is capable of providing state-of-the-art hashing-based video retrieval performances. More importantly, our BerVAE-based video hashing is the first model that is equipped with reasonable uncertainty quantification, which can help further calibrate the retrieval results for better decision making.

The current focus in this work is uncertainty-aware unsupervised video hashing. To further improve the retrieval performance, BerVAE can be modified with other learning strategies in the literature of learning to hash, such as contrastive learning and other supervised learning methods when additional video category labels are available, which we leave for our future research. Equipped with the uncertainty quantification capability, BerVAE-based video hashing has the promising potential to help advance content-based retrieval, especially for high-volume data such as videos.

There are several limitations of our current work. First, our current model is trained end to end in an purely unsupervised manner. In some scenarios where the supervision labels are easy to acquire, we may effectively and efficiently utilize those labels to improve the retrieval performance while preserving the uncertainty quantification capability, which we have not discussed in our current work. Second, as a simple illustration of the effectiveness of our quantified uncertainty, we gradually remove the query video clips with the uncertain hash-codes from the database. This strategy may not be always realistic in practice, and some advanced training and retrieval strategies based on the quantified uncertainty can be developed to further illustrate the importance of uncertainty quantification. We leave these for our future research.

Acknowledgements

This presented work is supported in part by the National Science Foundation (NSF) Awards: CCF-1553281, IIS-1812641, IIS-1812699, CCF-1934904, IIS-2212418, and IIS-2212419.

References

Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Broder, A. Z. (1997). On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.

Broder, A. Z., Glassman, S. C., Manasse, M. S., and Zweig, G. (1997). Syntactic clustering of the web. *Computer networks and ISDN systems*, 29(8-13):1157–1166.

Chaidaroon, S. and Fang, Y. (2017). Variational deep semantic hashing for text documents. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 75–84.

Dadaneh, S. Z., Boluki, S., Yin, M., Zhou, M., and Qian, X. (2020). Pairwise supervised hashing with bernoulli variational auto-encoder and self-control gradient estimator. In *Conference on Uncertainty in Artificial Intelligence*, pages 540–549. PMLR.

Dasgupta, A., Kumar, R., and Sarlós, T. (2011). Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1073–1081.

Dong, W., Su, Q., Shen, D., and Chen, C. (2019). Document hashing with mixture-prior generative models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5226–5235, Hong Kong, China. Association for Computational Linguistics.

Dong, Z., Mnih, A., and Tucker, G. (2020). Disarm: An antithetic gradient estimator for binary latent variables. In *Advances in neural information processing systems*, volume 33, pages 18637–18647.

Erin Liong, V., Lu, J., Wang, G., Moulin, P., and Zhou, J. (2015). Deep hashing for compact binary codes learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2475–2483.

Fajtl, J., Argyriou, V., Monekosso, D., and Remagnino, P. (2020). Latent bernoulli autoencoder. In *International Conference on Machine Learning*, pages 2964–2974. PMLR.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Heilbron, F. C., Escorcia, V., Ghanem, B., and Niebles, J. C. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970.

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613.

Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jiang, Y.-G., Wu, Z., Wang, J., Xue, X., and Chang, S.-F. (2018). Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):352–364.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Li, S., Chen, Z., Li, X., Lu, J., and Zhou, J. (2019a). Unsupervised variational video hashing with 1d-cnn-lstm networks. *IEEE Transactions on Multimedia*, 22(6):1542–1554.

Li, S., Chen, Z., Lu, J., Li, X., and Zhou, J. (2019b). Neighborhood preserving hashing for scalable video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8212–8221.

Li, S., Li, X., Lu, J., and Zhou, J. (2021). Self-supervised video hashing via bidirectional transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13549–13558.

Li, W.-J., Wang, S., and Kang, W.-C. (2016). Feature learning based deep supervised hashing with pairwise labels. In *Proceedings of the Twenty-Fifth International Joint*

Conference on Artificial Intelligence, IJCAI'16, page 1711–1717. AAAI Press.

Liu, W., Mu, C., Kumar, S., and Chang, S.-F. (2014). Discrete graph hashing. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Liu, W., Wang, J., Kumar, S., and Chang, S.-F. (2011). Hashing with graphs. In *International Conference on Machine Learning*.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Mena, F. and Ñanculef, R. (2019). A binary variational autoencoder for hashing. In *Iberoamerican Congress on Pattern Recognition*, pages 131–141. Springer.

Park, Y., Lee, S., Kim, G., and Blei, D. M. (2021). Unsupervised representation learning via neural activation coding. *CoRR*, abs/2112.04014.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library.

Razavi, A., van den Oord, A., and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Rolfe, J. T. (2016). Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

Salakhutdinov, R. and Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Song, J., Zhang, H., Li, X., Gao, L., Wang, M., and Hong, R. (2018). Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Transactions on Image Processing*, 27(7):3210–3221.

Strecha, C., Bronstein, A., Bronstein, M., and Fua, P. (2011). Ldahash: Improved matching with smaller descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):66–78.

The Theano Development Team, Al-Rfou, R., Alain, G., Almahairi, A., et al. (2016). Theano: A python framework for fast computation of mathematical expressions.

van den Oord, A., Vinyals, O., and kavukcuoglu, k. (2017). Neural discrete representation learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Verwilst, M., Žižakić, N., Gu, L., and Pižurica, A. (2021). Deep image hashing based on twin-bottleneck hashing with variational autoencoders. In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE.

Weiss, Y., Torralba, A., and Fergus, R. (2008). Spectral hashing. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc.

Yin, M., Ho, N., Yan, B., Qian, X., and Zhou, M. (2020). Probabilistic best subset selection via gradient-based optimization. *arXiv preprint arXiv:2006.06448*.

Yin, M. and Zhou, M. (2019). ARM: Augment-REINFORCE-merge gradient for stochastic binary networks. In *International Conference on Learning Representations*.

Yuan, L., Wang, T., Zhang, X., Tay, F. E., Jie, Z., Liu, W., and Feng, J. (2020). Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3083–3092.

Zhang, H., Wang, M., Hong, R., and Chua, T.-S. (2016). Play and rewind: Optimizing binary representations of videos by self-supervised temporal hashing. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 781–790.

A DERIVATION OF CLOSED-FORM GRADIENT W.R.T PREDICTED BERNOULLI SUCCESS PROBABILITIES

In this section, we provide the derivation of the closed-form gradient (CFG) of ELBO given the training video data. Our adopted minimization objective has the following expression:

$$\mathcal{L}_{ELBO}(\phi,\psi)=\frac{\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\sum_{i=1}^{N}\sum_{m=1}^{M}||\boldsymbol{v}_i^m-\boldsymbol{\mu}_i^m||_2^2]}{MNd}+\frac{\lambda\sum_{i=1}^{N}D_{KL}(q_\phi(\boldsymbol{b}_i|V_i)||p(\boldsymbol{b}_i))}{KN}. \tag{11}$$

The KL term is computed as $D_{KL}(q_\phi(\boldsymbol{b}_i|V_i)||p(\boldsymbol{b}_i))=\sum_{v=1}^{k}[\sigma(\bar{\boldsymbol{t}}_i^v)\log\frac{\sigma(\bar{\boldsymbol{t}}_i^v)}{\boldsymbol{p}^v}+(1-\sigma(\bar{\boldsymbol{t}}_i^v))\log\frac{1-\sigma(\bar{\boldsymbol{t}}_i^v)}{1-\boldsymbol{p}^v}]$, whose contributed gradient can be obtained by algebraic manipulations. Here we derive the closed-form gradient contributed by the likelihood term. The closed-form gradient of ELBO is just the summation of gradients contributed by two terms.

Given the parameters of the decoder network $\psi=\{\boldsymbol{w}_{\text{Rc1}},\boldsymbol{b}_{\text{Rc1}},W_{\text{Rc2}},\boldsymbol{b}_{\text{Rc2}}\}$, the mean of the modeled Gaussian generative distribution can be written: $\boldsymbol{\mu}_i=\boldsymbol{w}_{\text{Rc1}}\boldsymbol{b}_i^TW_{\text{Rc2}}$. The squared error L_v in the likelihood function of V_i has the following expression:

$$\begin{aligned}
\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\sum_{i=1}^{N}\sum_{m=1}^{M}\frac{1}{2\sigma_{\text{Rc}}^2}||\boldsymbol{v}_i^m-\boldsymbol{\mu}_i^m||_2^2]&=\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\sum_{i=1}^{N}\frac{1}{2\sigma_{\text{Rc}}^2}||V_i-\boldsymbol{\mu}_i||_2^2]\\
&=\sum_{i=1}^{N}\frac{1}{2\sigma_{\text{Rc}}^2}\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[||V_i-\boldsymbol{\mu}_i||_2^2]\\
&=\sum_{i=1}^{N}\frac{1}{2\sigma_{\text{Rc}}^2}\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\text{tr}(V_i^TV_i+\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i-2V_i^T\boldsymbol{\mu}_i)]\\
&=\sum_{i=1}^{N}\frac{1}{2\sigma_{\text{Rc}}^2}\text{tr}(\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[V_i^TV_i+\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i-2V_i^T\boldsymbol{\mu}_i])\\
&=\sum_{i=1}^{N}\frac{1}{2\sigma_{\text{Rc}}^2}\text{tr}(\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i-2V_i^T\boldsymbol{\mu}_i])+\text{tr}(V_i^TV_i),
\end{aligned} \tag{12}$$

where $\text{tr}(V_i^TV_i)$ is a constant with respect to the encoder parameters and therefore will not contribute to the computation of the gradient. The first trace term $\text{tr}(\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i])$ has the following expression:

$$\begin{aligned}
\text{tr}(\mathbb{E}[\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i])&=\text{tr}[\mathbb{E}[W_{\text{Rc2}}^T\boldsymbol{b}_i\boldsymbol{w}_{\text{Rc1}}^T\boldsymbol{w}_{\text{Rc1}}\boldsymbol{b}_i^TW_{\text{Rc2}}]]=(\boldsymbol{w}_{\text{Rc1}}^T\boldsymbol{w}_{\text{Rc1}})\text{tr}[W_{\text{Rc2}}^T\mathbb{E}[\boldsymbol{b}_i\boldsymbol{b}_i^T]W_{\text{Rc2}}]\\
&=(\boldsymbol{w}_{\text{Rc1}}^T\boldsymbol{w}_{\text{Rc1}})\text{tr}[W_{\text{Rc2}}W_{\text{Rc2}}^T\mathbb{E}[\boldsymbol{b}_i\boldsymbol{b}_i^T]].
\end{aligned} \tag{13}$$

The second part of the trace term $\text{tr}(\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[V_i^T\boldsymbol{\mu}_i])$ can be written as:

$$\text{tr}(\mathbb{E}[V_i^T\boldsymbol{\mu}_i])=\text{tr}[\mathbb{E}[V_i^T(\boldsymbol{w}_{\text{Rc1}}\boldsymbol{b}_i^TW_{\text{Rc2}})]]=\text{tr}[V_i^T\boldsymbol{w}_{\text{Rc1}}\mathbb{E}[(\boldsymbol{b}_i^T)]W_{\text{Rc2}}]. \tag{14}$$

Putting the previous equations (13) and (14) back into (12), the squared error L_v in the likelihood function of V_i can be expressed as:

$$\begin{aligned}
\mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\sum_{i=1}^{N}\sum_{m=1}^{M}\frac{1}{2\sigma_{\text{Rc}}^2}||\boldsymbol{v}_i^m-\boldsymbol{\mu}_i^m||_2^2]=\sum_{i=1}^{N}\frac{1}{2\sigma_{\text{Rc}}^2}&((\boldsymbol{w}_{\text{Rc1}}^T\boldsymbol{w}_{\text{Rc1}})\text{tr}[W_{\text{Rc2}}W_{\text{Rc2}}^T\mathbb{E}[\boldsymbol{b}_i\boldsymbol{b}_i^T]]\\
&-2\text{tr}[V_i^T\boldsymbol{w}_{\text{Rc1}}\mathbb{E}[(\boldsymbol{b}_i^T)]W_{\text{Rc2}}])+\text{tr}(V_i^TV_i).
\end{aligned} \tag{15}$$

The closed-form gradient of the squared error L_v with respect to each entry of the predicted Bernoulli success probabilities,

$\frac{\partial L_v}{\partial \sigma(\bar{\boldsymbol{t}}^v)}$, can be expressed as:

$$\begin{aligned}\frac{\partial L_v}{\partial \sigma(\bar{\boldsymbol{t}}^v)} &= \frac{\partial \mathbb{E}_{q_\phi(\mathcal{B}|\boldsymbol{S})}[\sum_{i=1}^N \sum_{m=1}^M \frac{1}{2\sigma_{\text{Rc}}^2}||\boldsymbol{v}_i^m - \boldsymbol{\mu}_i^m||_2^2]}{\partial \sigma(\bar{\boldsymbol{t}}^v)} \\ &= \frac{\partial \sum_{i=1}^N \frac{1}{2\sigma_{\text{Rc}}^2}((\boldsymbol{w}_{\text{Rc1}}^T \boldsymbol{w}_{\text{Rc1}})\text{tr}[W_{\text{Rc2}} W_{\text{Rc2}}^T \mathbb{E}[\boldsymbol{b}_i \boldsymbol{b}_i^T]] - 2\text{tr}[V_i^T \boldsymbol{w}_{\text{Rc1}} \mathbb{E}[(\boldsymbol{b}_i^T)] W_{\text{Rc2}}])}{\partial \sigma(\bar{\boldsymbol{t}}^v)} \\ &= \sum_{i=1}^N \frac{1}{2\sigma_{\text{Rc}}^2}((\boldsymbol{w}_{\text{Rc1}}^T \boldsymbol{w}_{\text{Rc1}})\text{tr}[W_{\text{Rc2}} W_{\text{Rc2}}^T \frac{\partial \mathbb{E}[\boldsymbol{b}_i \boldsymbol{b}_i^T]}{\partial \sigma(\bar{\boldsymbol{t}}^v)}] - 2\text{tr}[V_i^T \boldsymbol{w}_{\text{Rc1}} \frac{\partial \mathbb{E}[(\boldsymbol{b}_i^T)]}{\partial \sigma(\bar{\boldsymbol{t}}^v)} W_{\text{Rc2}}]).\end{aligned} \tag{16}$$

When we model the variational distribution $q_\phi(\boldsymbol{b}_i|V_i)$ to be a Bernoulli distribution with the success probability $\sigma(\bar{\boldsymbol{t}}^v)$ to be 1 and otherwise $1 - \sigma(\bar{\boldsymbol{t}}^v)$ to be -1, $\frac{\partial \mathbb{E}[\boldsymbol{b}_i \boldsymbol{b}_i^T]}{\partial \sigma(\bar{\boldsymbol{t}}^v)}$ has the following expression:

$$\frac{\partial \mathbb{E}[\boldsymbol{b}_i \boldsymbol{b}_i^T]}{\partial \sigma(\bar{\boldsymbol{t}}^v)} = \begin{bmatrix} 0 & 0 & \dots & 4\sigma(\bar{\boldsymbol{t}}^1) - 2 & \dots & 0 & 0 \\ 0 & 0 & \dots & \vdots & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 4\sigma(\bar{\boldsymbol{t}}^{v-1}) - 2 & \vdots & \vdots & \\ 4\sigma(\bar{\boldsymbol{t}}^1) - 2 & \dots & 4\sigma(\bar{\boldsymbol{t}}^{v-1}) - 2 & 0 & 4\sigma(\bar{\boldsymbol{t}}^{v+1}) - 2 & \dots & 4\sigma(\bar{\boldsymbol{t}}^k) - 2 \\ \vdots & \vdots & \vdots & 4\sigma(\bar{\boldsymbol{t}}^{v+1}) - 2 & \ddots & \vdots & \\ 0 & 0 & \dots & \vdots & \dots & 0 & 0 \\ 0 & 0 & \dots & 4\sigma(\bar{\boldsymbol{t}}^k) - 2 & \dots & 0 & 0 \end{bmatrix}, \tag{17}$$

and $\frac{\partial \mathbb{E}[(\boldsymbol{b}_i^T)]}{\partial \sigma(\bar{\boldsymbol{t}}^v)} = (0, 0, \dots, 2\sigma(\bar{\boldsymbol{t}}^v), \dots, 0, 0)$ with the n-th entry to be $2\sigma(\bar{\boldsymbol{t}}^v)$.

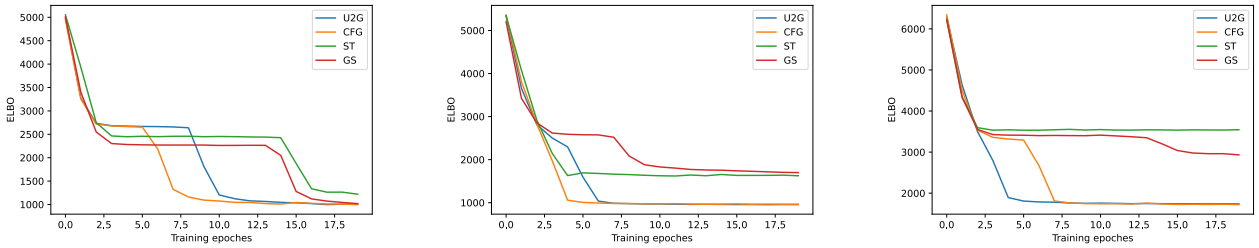
B SUPPLEMENTARY EXPERIMENTAL RESULTS ON TOY EXAMPLES

In this section, we include more results of another toy example. Similar as in the reported experiments for the toy example in Section 4.1 in the main text, both of the training and test sets here are composed of 10,000 points independently sampled from a mixture of 2-D Gaussian distributions with the mean $\boldsymbol{\mu}_1 = [-0.5, 0]^T, \boldsymbol{\mu}_2 = [0.5, 0]^T$ and $\boldsymbol{\mu}_3 = [0, 1]^T$ and variance $\sigma^2 I$ with the same σ values as in Section 4.1 along with another case of $\sigma = 0.2$. We include the training ELBO in Figure 12. As the y-coordinates of the Gaussian mixture component centers are no longer the same, here we plot the predicted uncertainty using the heatmap together with the hash-code latent representations in Figure 13. The lighter color of the heatmap represents higher predicted uncertainty of the latent hash-codes around while the darker regions represent lower uncertainty of the hash-codes nearby. The points encoded into 2-bit hash-codes $\{-1, -1\}, \{-1, 1\}, \{1, -1\}$ and $\{1, 1\}$ are represented in blue, green, cyan and magenta, respectively.

In Figure 12, we can see clearly that the unbiased CFG and U2G estimators achieve consistently the better training loss than the biased ST and GS estimators. With enough training epochs, using the U2G estimator performs similarly as training by CFG. The performance differences are also confirmed in the clustering and uncertainty quantification results in Figure 13. All four gradient estimators can perform well in the easiest case with $\sigma = 0.15$. As the variance goes higher and three mixture components have larger overlap, it is more challenging for the BerVAE-based hashing model to reconstruct all three modes. With $\sigma = 0.2$, the models trained by the unbiased U2G and CFG can still maintain satisfactory clustering results and reasonable uncertainty quantification, both significantly outperforming ST and GS estimators. In the third case with $\sigma = 0.3$, where the boundaries between three modes are even more ambiguous, the BerVAE models trained with U2G and CFG can still roughly reconstruct the modes while the model trained with the biased ST estimator fails to reconstruct all three different modes and overestimates the uncertainty. All the above results and discussion support our choice of using the U2G estimator in our final BerVAE training implementation.

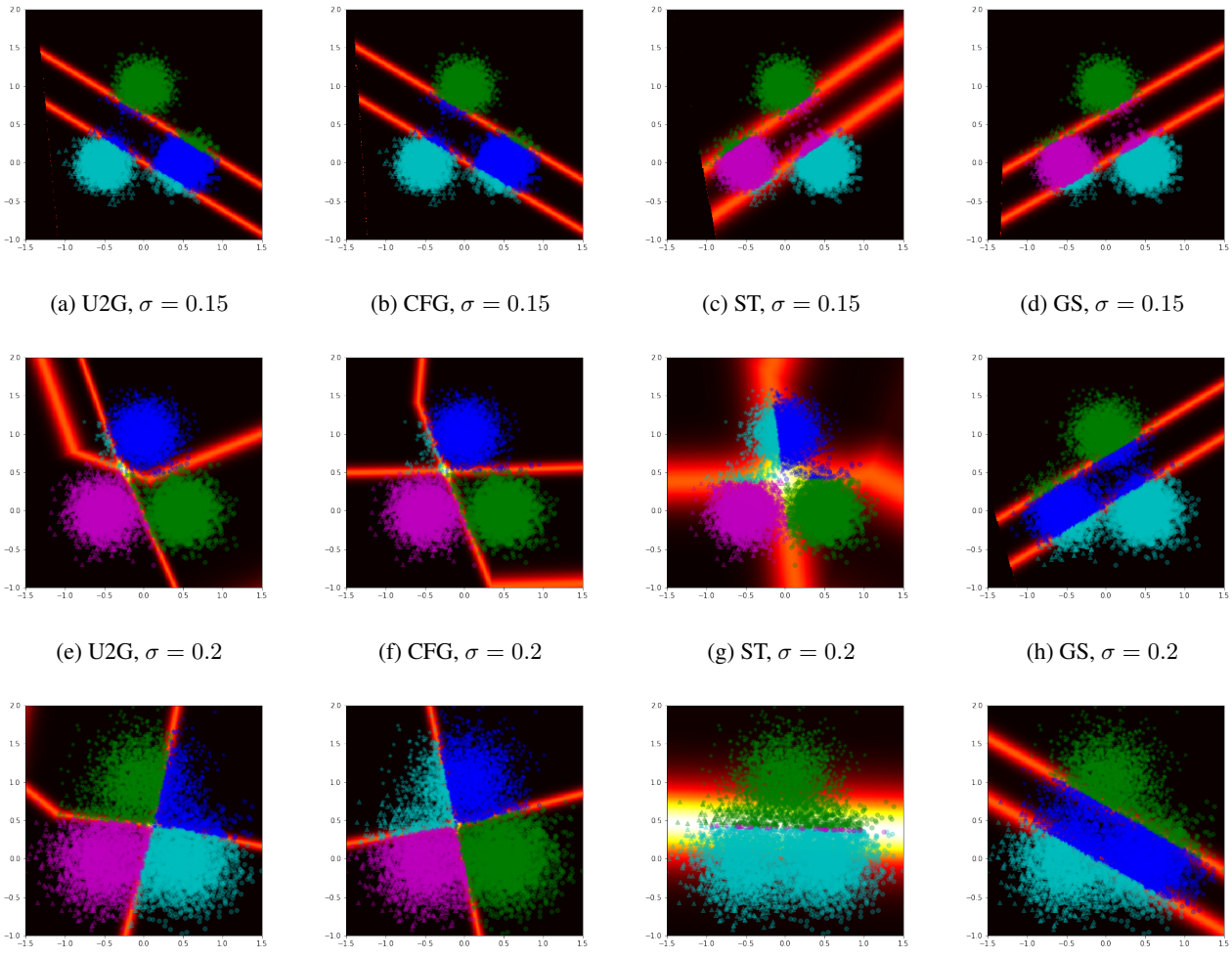
C SUPPLEMENTARY EXPERIMENTAL RESULTS ON VIDEO DATASETS

In this section, we provide the additional experimental results on real-world large-scale video datasets. We first provide the additional experimental results on FCVID in Section C.1. Then we present our experiments performed on the ActivityNet dataset in Section C.2. In Section C.3, we also include the ablation experiments studying the effect of hyperparameter λ.



(a) $\sigma = 0.15$ (b) $\sigma = 0.2$ (c) $\sigma = 0.3$

Figure 12: Training ELBO with respect to the training epochs of the BerVAE trained with different gradient estimators.

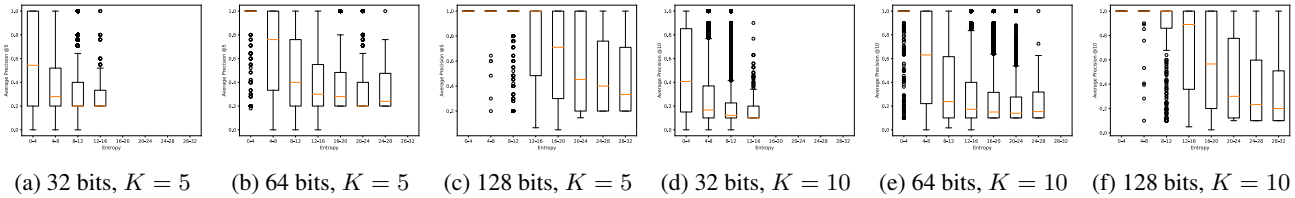


(a) U2G, $\sigma = 0.15$ (b) CFG, $\sigma = 0.15$ (c) ST, $\sigma = 0.15$ (d) GS, $\sigma = 0.15$

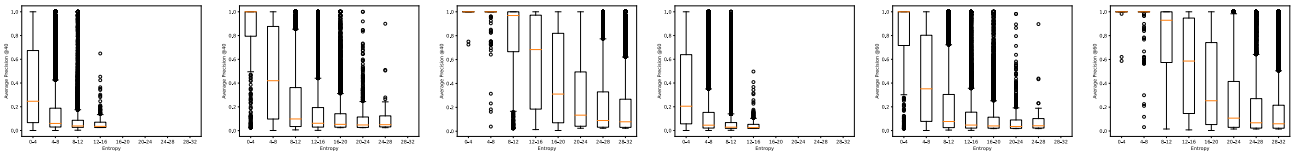
(e) U2G, $\sigma = 0.2$ (f) CFG, $\sigma = 0.2$ (g) ST, $\sigma = 0.2$ (h) GS, $\sigma = 0.2$

(i) U2G, $\sigma = 0.3$ (j) CFG, $\sigma = 0.3$ (k) ST, $\sigma = 0.3$ (l) GS, $\sigma = 0.3$

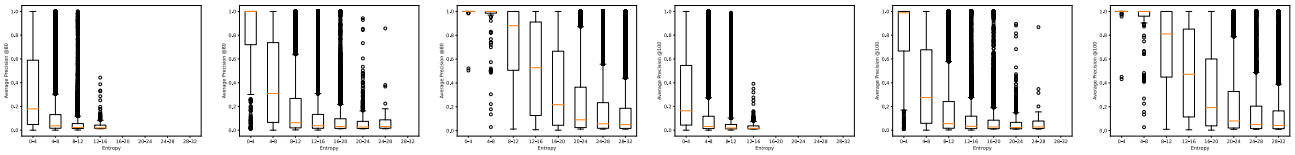
Figure 13: The clustering results and predicted uncertainty by BerVAE trained with different gradient estimators.



(a) 32 bits, $K = 5$ (b) 64 bits, $K = 5$ (c) 128 bits, $K = 5$ (d) 32 bits, $K = 10$ (e) 64 bits, $K = 10$ (f) 128 bits, $K = 10$



(g) 32 bits, $K = 40$ (h) 64 bits, $K = 40$ (i) 128 bits, $K = 40$ (j) 32 bits, $K = 60$ (k) 64 bits, $K = 60$ (l) 128 bits, $K = 60$



(m) 32 bits, $K = 80$ (n) 64 bits, $K = 80$ (o) 128 bits, $K = 80$ (p) 32 bits, $K = 100$ (q) 64 bits, $K = 100$ (r) 128 bits, $K = 100$

Figure 14: Box-plots of AP@K values with respect to the predicted uncertainty of video hash-codes with $K = 5, 10, 40, 60, 80, 100$ on FCVID.

C.1 Additional Results on FCVID

We include the box-plots of the AP@K values of the query video clips with respect to the uncertainty level of corresponding hash-codes of the queries with $K = 5, 10, 40, 60, 80, 100$ in Figure 14. All the results with different bit-lengths of hash-codes have demonstrated that as the hash-code uncertainty goes higher, both the medians and quantiles of the distributions of query video clips' AP values become smaller, similar as the reported results of $K = 20$ in Section 4.6 in the main text. Again, it is clear that the estimated uncertainty level by BerVAE has significant correlation with video retrieval performance.

We have also reported the quantitative mAP values of BerVAE and the baseline models on FCVID dataset in Tables 1, 2, 3, as well as the quantitative IDU values in Tables 7. The rows marked as 'BerVAE+Nei' in these tables present the retrieval results of our BerVAE trained with the combined contrastive loss term as the one adopted in Li et al. (2021). Our BerVAE maintains consistent performance advantage over the BTH with 64- and 128-bit hash-codes with the combined loss terms. Moreover, the uncertainty quantification capability of our BerVAE is still well preserved.

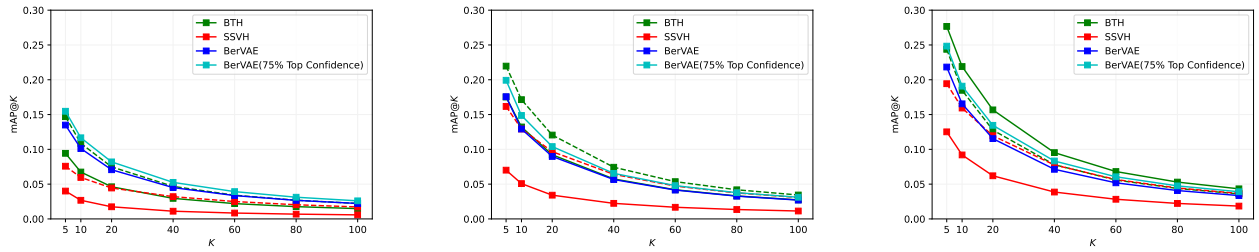
C.2 Experiments on ActivityNet

ActivityNet (Heilbron et al., 2015) is a large-scale human action recognition dataset. It contains videos from 203 activity classes with a total of 849 hours. We follow the previous work (Li et al., 2021) and use 9,722 videos for training, 1,000 and 3,760 videos from the validation subset of the official division for queries and retrievals.

Following (Li et al., 2021), we randomly select $M = 30$ frames for each video in ActivityNet and the dimension of the feature representation of each frame is 2,048. We set the prior strength λ in the training loss for our BerVAE to be 0.01. We first pretrain our BerVAE with the ST estimator for 50 epochs and then U2G for the remaining 150 epochs. All the network architecture settings and the other hyperparameters are kept to be the same as those implemented on FCVID.

C.2.1 Retrieval Accuracy Compared with Baseline Models on ActivityNet

Similar as the experiments performed on FCVID, we report the retrieval performance of our BerVAE along with BTH (Li et al., 2021) and SSVH (Song et al., 2018) on ActivityNet (Heilbron et al., 2015) in Figure 15. The performance trends of the models trained with combined losses and the reconstruction loss only are plotted in dotted lines and solid lines respectively. For fair comparison, we focus on the performance of our BerVAE with the models trained with only the



(a) ActivityNet, 32 bits (b) ActivityNet, 64 bits (c) ActivityNet, 128 bits

Figure 15: Retrieval accuracy of BerVAE compared with state-of-the-art models with different bit length of hash-codes on ActivityNet. K is the number of retrieved video clips when calculating mAP@K given a query video.

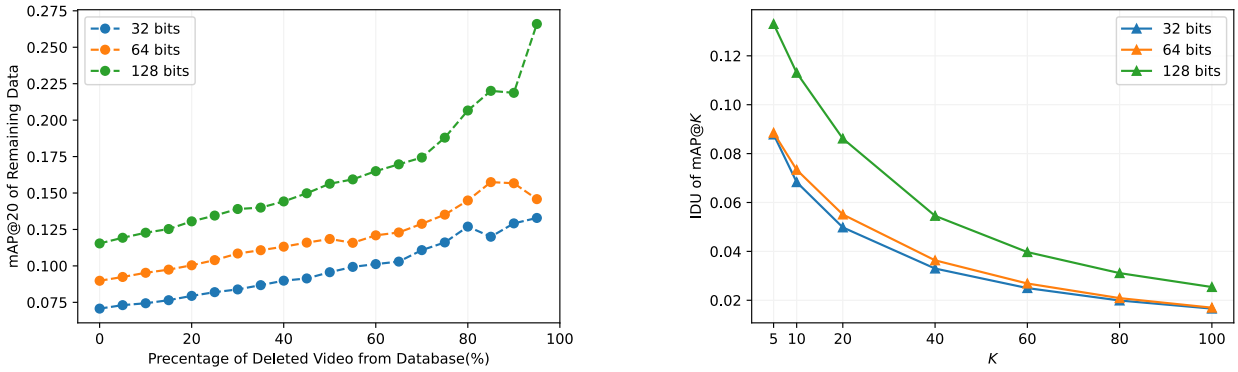
reconstruction loss. Compared to SSVH, our BerVAE achieves consistently better retrieval accuracy regardless of the hash-code bit-length and training loss. Our BerVAE also achieves 0.025 better mAP@20 with 32-bit hash-codes and similar performance with 64-bit hash-codes compared to BTH. To achieve comparable performance with BTH trained with combined losses, we need to withheld approximately 40% query clips with 128-bit hash-codes and 60% query clips with 64-bit hash-codes. The quantitative mAP results of BerVAE and the baseline models on ActivityNet dataset are included in Table 4, 5, and 6.

C.2.2 Uncertainty Quantification on ActivityNet

As uncertainty-aware video retrieval is the focus of this work, we have also studied the uncertainty quantification capability of our BerVAE on ActivityNet. As the query video clips does not overlap with the retrieval videos, here we report mAP@20 values for the remaining query video clips when we withhold specific percentages of query videos with uncertain hash-codes in Figure 16a while we retrieve all the similar video clips based on the Hamming distances of the derived hash-codes. As more and more query videos with uncertain predicted hash-codes being withheld, the average retrieval accuracy of the remaining data gets significantly better, similar as on the FCVID dataset. In Figure 16b, we also report the IDU of mAP@K with different K's. With different hash-code bit-length, our BerVAE can stably achieve reasonable uncertainty quantification. Both the retrieval accuracy and uncertainty quantification performances on ActivityNet are worse than the obtained results on FCVID, especially for large K's. A possible explanation is that the sample size of ActivityNet is significantly smaller than FCVID. With insufficient training data, the model can not learn a good latent distribution. Moreover, ActivityNet contains more action-related video categories which is hard to cluster automatically in the current unsupervised setting. The quantitative IDU values of our BerVAE are included in Table 8.

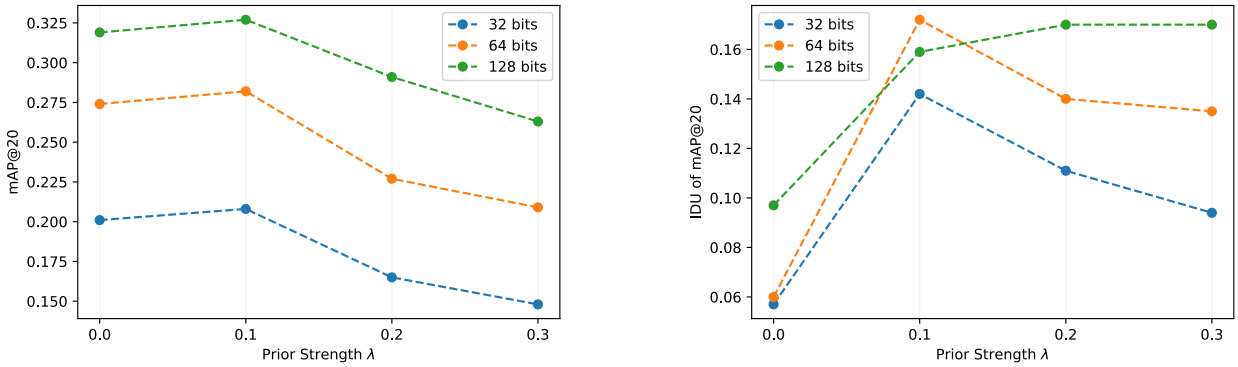
C.3 Ablation Study on the Effect of Prior Strength λ on FCVID

We study the effect of λ, which represents our prior belief relative to collected data, on both retrieval performance and uncertainty quantification by performing ablation experiments on FCVID. We report mAP@20 and IDU of mAP@20 with different λ values in Figures 17a and 17b, respectively. We achieve consistently the best retrieval and uncertainty quantification performance with the λ set to be 0.1 when the learned hash-codes are 32- and 64-bits. The achieved retrieval accuracy of 128-bit hash-codes with $\lambda = 0.1$ is the best among all the setups, which may account for its degraded uncertainty quantification performance. With the KL term removed from the training loss, which is equivalent to setting $\lambda = 0$, the model can hardly quantify the uncertainty of derived hash-codes reasonably, which demonstrates the importance of the prior in terms of the model uncertainty quantification.



(a) (b)

Figure 16: (a) mAP@20 of remaining video clips with respect to the percentage of uncertain videos withheld from queries on ActivityNet; (b) IDU of mAP@K with different K on ActivityNet.



(a) (b)

Figure 17: Ablation study with respect to (w.r.t.) the training loss coefficient λ on FCVID. (a) Retrieval accuracy w.r.t. the prior strength λ. (b) Uncertainty quantification performance w.r.t. the prior strength λ.

	mAP@K						
Model	$K=5$	$K=10$	$k=20$	$K=40$	$K=60$	$K=80$	$K=100$
BTH(Reported)	-	-	0.248	0.204	0.182	0.166	0.154
NPH(Reported)	-	-	0.246	0.195	0.170	0.154	0.141
BTH(Reproduced)	0.381	0.284	0.222	0.178	0.157	0.143	0.133
SSVH	0.384	0.274	0.204	0.158	0.137	0.124	0.115
BTH(Rec. loss)	0.377	0.265	0.194	0.147	0.125	0.112	0.102
SSVH(Rec. loss)	0.344	0.233	0.164	0.119	0.099	0.086	0.078
BerVAE	0.388	0.278	0.208	0.160	0.138	0.125	0.115
BerVAE(25% withheld)	0.413	0.304	0.233	0.183	0.160	0.146	0.135
BerVAE(50% withheld)	0.443	0.338	0.266	0.215	0.191	0.175	0.163
BerVAE(75% withheld)	0.500	0.401	0.333	0.279	0.251	0.231	0.216
BerVAE+Nei	0.370	0.270	0.206	0.162	0.142	0.129	0.119
BerVAE+Nei(25% withheld)	0.400	0.304	0.239	0.194	0.172	0.158	0.147
BerVAE+Nei(50% withheld)	0.437	0.348	0.286	0.239	0.215	0.198	0.185
BerVAE+Nei(75% withheld)	0.495	0.410	0.350	0.299	0.272	0.253	0.238

Table 1: Retrieval accuracy by BerVAE compared with different state-of-the-art models with 32-bit hash-codes on FCVID.

	mAP@K						
Model	$K = 5$	$K = 10$	$k = 20$	$K = 40$	$K = 60$	$K = 80$	$K = 100$
BTH(Reported)	-	-	0.308	0.260	0.234	0.216	0.202
NPH(Reported)	-	-	0.294	0.240	0.213	0.196	0.183
BTH(Reproduced)	0.465	0.366	0.300	0.252	0.227	0.210	0.196
SSVH	0.464	0.355	0.282	0.228	0.202	0.185	0.172
BTH(Rec. loss)	0.450	0.338	0.262	0.207	0.181	0.163	0.150
SSVH(Rec. loss)	0.432	0.317	0.239	0.183	0.157	0.140	0.127
BerVAE	0.467	0.357	0.282	0.226	0.199	0.181	0.167
BerVAE(25% withheld)	0.493	0.385	0.310	0.254	0.225	0.205	0.190
BerVAE(50% withheld)	0.527	0.424	0.350	0.292	0.261	0.238	0.221
BerVAE(75% withheld)	0.594	0.501	0.431	0.371	0.334	0.305	0.282
BerVAE+Nei	0.484	0.382	0.313	0.262	0.235	0.216	0.201
BerVAE+Nei(25% withheld)	0.531	0.433	0.366	0.312	0.283	0.262	0.245
BerVAE+Nei(50% withheld)	0.590	0.500	0.434	0.378	0.346	0.321	0.301
BerVAE+Nei(75% withheld)	0.683	0.607	0.544	0.484	0.446	0.416	0.389

Table 2: Retrieval accuracy by BerVAE compared with different state-of-the-art models with 64-bit hash-codes on FCVID.

	mAP@K						
Model	$K = 5$	$K = 10$	$k = 20$	$K = 40$	$K = 60$	$K = 80$	$K = 100$
BTH(Reported)	-	-	-	-	-	-	-
NPH(Reported)	-	-	-	-	-	-	-
BTH(Reproduced)	0.510	0.415	0.350	0.301	0.274	0.255	0.239
SSVH	0.516	0.413	0.341	0.286	0.258	0.238	0.222
BTH(Rec. loss)	0.500	0.390	0.312	0.252	0.221	0.201	0.185
SSVH(Rec. loss)	0.477	0.361	0.280	0.219	0.189	0.169	0.154
BerVAE	0.513	0.405	0.327	0.266	0.234	0.213	0.196
BerVAE(25% withheld)	0.533	0.428	0.351	0.290	0.257	0.234	0.216
BerVAE(50% withheld)	0.563	0.461	0.386	0.323	0.289	0.264	0.244
BerVAE(75% withheld)	0.629	0.537	0.466	0.399	0.359	0.327	0.299
BerVAE+Nei	0.530	0.432	0.365	0.312	0.283	0.262	0.244
BerVAE+Nei(25% withheld)	0.578	0.487	0.421	0.367	0.335	0.311	0.291
BerVAE+Nei(50% withheld)	0.640	0.559	0.496	0.439	0.403	0.375	0.352
BerVAE+Nei(75% withheld)	0.731	0.664	0.606	0.542	0.499	0.462	0.429

Table 3: Retrieval accuracy by BerVAE compared with different state-of-the-art models with 128-bit hash-codes on FCVID.

	mAP@K						
Model	$K = 5$	$K = 10$	$k = 20$	$K = 40$	$K = 60$	$K = 80$	$K = 100$
BTH(Reported)	-	-	-	-	-	-	-
BTH(Reproduced)	0.147	0.109	0.075	0.047	0.034	0.027	0.022
SSVH(Reproduced)	0.076	0.060	0.044	0.032	0.025	0.020	0.017
BTH(Rec. loss)	0.094	0.067	0.046	0.030	0.022	0.017	0.015
SSVH(Rec. loss)	0.040	0.027	0.017	0.011	0.008	0.007	0.006
BerVAE	0.135	0.101	0.071	0.045	0.034	0.0267	0.022
BerVAE(25% withheld)	0.155	0.117	0.082	0.053	0.039	0.031	0.026
BerVAE(50% withheld)	0.176	0.134	0.096	0.061	0.046	0.036	0.030
BerVAE(75% withheld)	0.212	0.163	0.116	0.075	0.057	0.045	0.038

Table 4: Retrieval accuracy by BerVAE compared with different state-of-the-art models with 32-bit hash-codes on ActivityNet.

	mAP@K						
Model	$K = 5$	$K = 10$	$k = 20$	$K = 40$	$K = 60$	$K = 80$	$K = 100$
BTH(Reported)	-	-	-	-	-	-	-
BTH(Reproduced)	0.244	0.171	0.121	0.074	0.054	0.042	0.034
SSVH(Reproduced)	0.162	0.129	0.097	0.064	0.047	0.037	0.031
BTH(Rec. loss)	0.175	0.132	0.092	0.057	0.042	0.033	0.027
SSVH(Rec. loss)	0.070	0.051	0.034	0.022	0.017	0.014	0.011
BerVAE	0.176	0.129	0.090	0.057	0.041	0.033	0.027
BerVAE(25% withheld)	0.199	0.149	0.104	0.065	0.048	0.038	0.031
BerVAE(50% withheld)	0.221	0.168	0.118	0.076	0.055	0.044	0.036
BerVAE(75% withheld)	0.249	0.190	0.135	0.087	0.064	0.050	0.041

Table 5: Retrieval accuracy by BerVAE compared with different state-of-the-art models with 64-bit hash-codes on ActivityNet.

	mAP@K						
Model	$K = 5$	$K = 10$	$k = 20$	$K = 40$	$K = 60$	$K = 80$	$K = 100$
BTH(Reported)	-	-	-	-	-	-	-
BTH(Reproduced)	0.277	0.219	0.157	0.095	0.068	0.053	0.043
SSVH(Reproduced)	0.194	0.159	0.120	0.077	0.057	0.044	0.037
BTH(Rec. loss)	0.244	0.185	0.128	0.078	0.056	0.044	0.036
SSVH(Rec. loss)	0.125	0.092	0.062	0.039	0.028	0.022	0.018
BerVAE	0.218	0.165	0.115	0.071	0.052	0.041	0.034
BerVAE(25% withheld)	0.248	0.191	0.135	0.083	0.061	0.047	0.039
BerVAE(50% withheld)	0.285	0.221	0.156	0.096	0.070	0.055	0.045
BerVAE(75% withheld)	0.333	0.262	0.188	0.117	0.085	0.067	0.055

Table 6: Retrieval accuracy by BerVAE compared with different state-of-the-art models with 128-bit hash-codes on ActivityNet.

	IDU of mAP@K						
Length	$K = 5$	$K = 10$	$k = 20$	$K = 40$	$K = 60$	$K = 80$	$K = 100$
32-bit	0.124	0.139	0.142	0.134	0.127	0.119	0.113
64-bit	0.148	0.168	0.172	0.163	0.154	0.143	0.134
128-bit	0.133	0.153	0.159	0.151	0.140	0.128	0.116

Table 7: IDU of video retrieval results using the derived hash-codes with different bit length by BerVAE on FCVID.

	IDU of mAP@K						
Length	$K = 5$	$K = 10$	$k = 20$	$K = 40$	$K = 60$	$K = 80$	$K = 100$
32-bit	0.088	0.068	0.050	0.033	0.025	0.020	0.017
64-bit	0.089	0.073	0.055	0.036	0.027	0.021	0.017
128-bit	0.133	0.113	0.086	0.054	0.040	0.031	0.025

Table 8: IDU of video retrieval results using the derived hash-codes with different bit length by BerVAE on ActivityNet.